ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO

FACULTAD DE INFORMÁTICA Y ELECTRÓNICA

ESCUELA DE INGENIERÍA ELECTRÓNICA EN TELECOMUNICACIONES Y REDES

"ANÁLISIS E IMPLEMENTACIÓN DE ALTA DISPONIBILIDAD MEDIANTE CLUSTERING EN SISTEMAS DE CALL CENTER BASADOS EN VoIP"

TESIS DE GRADO

PREVIA OBTENCIÓN DEL TÍTULO DE

INGENIERA EN ELECTRÓNICA TELECOMUNICACIONES Y REDES

PRESENTADO POR

SILVIA FABIOLA CUJANO OÑATE

RIOBAMBA – ECUADOR

2011

A mis padres, por el apoyo incondicional a lo largo de los años de estudio y sobre todo durante el desarrollo de esta investigación.

A los docentes de la Escuela de Electrónica por compartir sus conocimientos en las aulas.

A todos ellos, gracias.

Como muestra de cariño, esta investigación va dedicada a mis padres, los seres más importantes en mi vida, quienes día a día me han guiado para poder vencer los obstáculos y alcanzar esta meta.

A mis hermanos, familiares y amigos por sus consejos y palabras de apoyo, que me sirvieron de fuerza para no decaer.

NOMBRE	FIRMA	FECHA
Ing. Iván Menes **DECANO DE LA FACULTAD DE INFORMÁTICA Y ELECTRÓNICA**	_____	_____
Ing. Pedro Infante **DIRECTOR DE ESCUELA DE INGENIERÍA ELECTRÓNICA EN TELECOMUNICACIONES Y REDES**	_____	_____
Ing. Alberto Arellano **DIRECTOR DE TESIS**	_____	_____
Ing. José Guerra **MIEMBRO DEL TRIBUNAL**	_____	_____
Lcdo. Carlos Rodríguez **DIRECTOR DEL CENTRO DE DOCUMENTACIÓN**	_____	_____
NOTA DE LA TESIS	_____	

Yo, **Silvia Fabiola Cujano Oñate**, soy responsable de las ideas, doctrinas y resultados expuestos en esta Tesis y el patrimonio intelectual de la misma pertenece a la ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO.

Silvia Fabiola Cujano Oñate

ABREVIATURAS

ACD Automatic Call Distributor, Distribuidor Automático de Llamadas.

ATA Adaptador de Teléfono Analógico.

CC Call Center, Centro de Llamadas.

CIB Cluster Information Base, Base de Información del Clúster.

CRM Cluster Resource Manager, Administrador de Recursos del Clúster.

CTI Computer Telephone Integration, Integración Computadora-Teléfono.

ETSI European Telecommunications Standards Institute, Instituto Europeo de Normas de Telecomunicaciones.

FXO Foreign eXchange Station, Puerto que recibe la línea analógica.

FXS Foreign Exchange Office, Interfaz de abonado externo.

IETF Internet Engineering Task Force, grupo de Trabajo de Ingeniería de Internet.

ITU International Telecommunications Union, Unión Internacional de Telecomunicaciones.

IVR Interactive Voice Response, Respuesta Interactiva de Voz

LRM Local Resource Manager, Administrador de Recursos Locales

LSB Linux Standard Base,

OCF Open Cluster Framework,

PBX Private Branch Exchange, Central de Conmutación de llamadas telefónicas.

PSTN Public Switched Telephone Network, Red Telefónica Pública Conmutada.

RTP Real time Transport Protocol, Protocolo de Transporte en Tiempo Real.

SIP Session Initiation Protocol, Protocolo de Inicio de Sesión.

ÍNDICE GENERAL

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

INTRODUCCIÓN

En la actualidad el uso de Centros de Llamadas (Call Center) está dominado por grandes empresas que requieren establecer un contacto permanente con sus clientes, por tal razón existe la necesidad de que los servicios que brinda un CC sean proporcionados de manera ininterrumpida. Para conseguir estos niveles de disponibilidad se suele utilizar una configuración avanzada de hardware y de software denominada Clúster de Alta Disponibilidad.

Mediante clusters de alta disponibilidad se obtiene sistemas tolerantes a fallos, evitando de esta forma que fallas tanto de hardware como de software afecten el servicio que se brinda, es dentro de este entorno donde se presenta la solución, que consiste en la instalación y configuración de un clúster de alta disponibilidad cuyo sistema operativo puede ser Linux.

La presente investigación está dividida en cinco capítulos. El primer capítulo contiene las generalidades del proyecto de tesis, como los antecedentes, justificación para su realización, los objetivos a alcanzar y la hipótesis planteada.

El segundo capítulo cita la información de mayor importancia y que se cree es necesario conocerla para el entendimiento de los siguientes capítulos, comenzando con una introducción a lo que es VoIP, elementos, equipamiento, protocolos, codecs y los conceptos básicos involucrados en ambientes de Call Center.

En el tercer capítulo se empieza con una breve introducción a los tipos de clúster y sus componentes, para luego incluir conceptos de alta disponibilidad, así como también se habla de diferentes soluciones de Clúster de Alta Disponibilidad basadas en proyectos Open Source.

En el capítulo cuarto se incluye los requerimientos software y hardware para la implementación de un prototipo y se detalla las configuraciones realizadas tanto en el Módulo de Call Center como en el Clúster de Alta disponibilidad, además se mencionan los comandos mas utilizados en la administración del clúster.

Finalmente en el quinto capítulo se indica el funcionamiento del prototipo antes configurado, incluyendo también las pruebas a las cuales fue sometido el mismo, para comprobar su comportamiento.

CAPÍTULO I

MARCO REFERENCIAL

1.1. ANTECEDENTES

Debido a la globalización de los mercados los clientes se han vuelto más exigentes, las empresas reconocen que se pueden obtener ventajas competitivas sustanciales mediante un mejor servicio al cliente, y para esto deben conocer cuáles son las necesidades de su mercado y mantenerse siempre actualizadas a los cambios en este.

Es por estas razones que los Call Centers se han convertido en la mejor herramienta de negocios para satisfacer las exigentes necesidades de comunicación directa y personalizada entre empresa y cliente. Siendo así, existe la necesidad de que los servicios que brinda un CC sean proporcionados de manera ininterrumpida, 24 horas al día, 7 días a la semana. Para conseguir estos niveles de disponibilidad se suele utilizar una configuración avanzada de hardware y de software denominada Clúster de Alta Disponibilidad.

Un Clúster de Alta Disponibilidad es un conjunto de dos o más máquinas que se caracterizan porque comparten los discos de almacenamiento de datos, memoria

RAM, microprocesador y porque están constantemente monitorizadas entre sí. Si se produce un fallo de hardware o de las aplicaciones de alguna de las máquinas del clúster, el software de alta disponibilidad es capaz de arrancar automáticamente los servicios que han fallado en cualquiera de las otras máquinas del clúster. Y cuando la máquina que ha fallado se recupera, los servicios son nuevamente migrados a la máquina original.

Esta capacidad de recuperación automática de servicios nos garantiza la integridad de la información, ya que no hay pérdidas de datos y además evita molestias a los usuarios, que no tienen por qué notar que se ha producido un problema.

1.2. JUSTIFICACIÓN

Justificación Teórica

En la actualidad el uso de Centros de Llamadas está dominado por grandes empresas que requieren establecer un contacto permanente con sus clientes. En general compañías en el área de mercado masivo como empresas financieras, empresas de televenta, empresas gubernamentales (ej.: SRI, IESS), bancos (ej.: Banco del Pichincha, Banco de Guayaquil), servicios básicos y telecomunicaciones (ej.: Movistar, Claro).

Sin disminuir la importancia de la prestación de los servicios en cada uno de los ejemplos de Call Center antes mencionados, es aún más importante, cuando se habla de servicios de misión crítica, como por ejemplo hospitales o el servicio de 911, es así que en esos ambientes se hace indispensable contar con mecanismos que aseguren que el servicio sea brindado en el tiempo oportuno.

Mediante clústeres de alta disponibilidad se obtiene sistemas tolerantes a fallos, evitando de esta forma que fallas tanto de hardware como de software afecten el servicio que se brinda.

Es dentro de este entorno donde se presenta la solución, que consiste en la instalación y configuración de un clúster de alta disponibilidad bajo Linux.
Para la implementación de este clúster se podría utilizar Heartbeat como software de control y Debian GNU/Linux como sistema operativo.

En lo referente a monitorización del clúster existe variedad de software y para diferentes plataformas, en el presente caso podría utilizarse Ganglia.

Justificación Práctica

Los clúster de alta disponibilidad están diseñados para garantizar el funcionamiento ininterrumpido de ciertas aplicaciones, minimizando así la percepción del fallo por parte de los usuarios.

La utilización de clústeres no solo es beneficiosa para caídas de servicio no programadas, sino que también es útil en paradas de sistema programadas como puede ser un mantenimiento hardware o una actualización software.

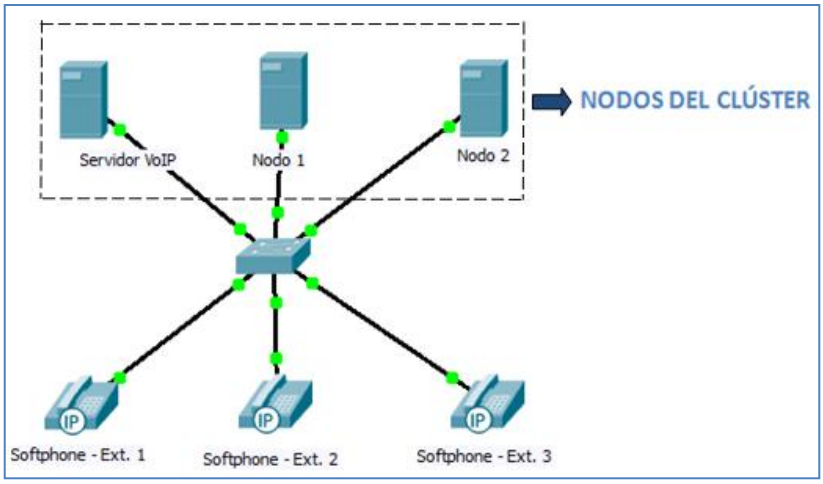

Figura I. 1 Diagrama del Prototipo

1.3. OBJETIVOS

1.3.1. OBJETIVO GENERAL

- ✓ Realizar el Análisis e Implementación de Alta Disponibilidad mediante Clustering en sistemas de Call Center basados en VoIP.

1.3.2. OBJETIVOS ESPECIFICOS

- ✓ Estudiar la arquitectura y aplicaciones de los sistemas de CALL CENTER (CC).
- ✓ Analizar las ventajas de la utilización de clúster en los servicios de CALL CENTER basados en VoIP.
- ✓ Implementar el prototipo de Clustering que permita alta disponibilidad.
- ✓ Realizar pruebas con el fin de comprobar la Alta Disponibilidad en el servicio.

1.4. HIPÓTESIS

El Análisis e Implementación de Alta Disponibilidad mediante Clustering permitirá disminuir los tiempos de no accesibilidad a los Sistemas de Call Center basados en VoIP.

CAPÍTULO II

INTRODUCCIÓN A VoIP

2.1. VoIP (Voice over IP)

La Voz sobre IP hace posible la transmisión de voz a través de redes IP en forma de paquetes de datos.

Debe quedar claro que VoIP no es un servicio como tal, sino una tecnología que usa el Protocolo de Internet (IP) a través de la cual se comprimen y descomprimen de manera altamente eficiente paquetes de datos o datagramas, para permitir la comunicación de dos o más clientes a través de una red como la red de Internet. Con esta tecnología pueden prestarse servicios de Telefonía o Videoconferencia, entre otros.

2.1.1. ELEMENTOS

Se definen tres elementos fundamentales en un ambiente de VoIP:

- ✓ **Terminales.-** Son los sustitutos de los actuales teléfonos. Se pueden implementar tanto en software (Softphone) como en hardware (Teléfonos IP).

- ✓ **Gatekeepers/Servidores.-** Son el centro de toda la organización VoIP, y son el sustituto para las actuales centrales. Normalmente implementadas por software.

  Estos elementos se encargan de realizar operaciones tales como, administración y control del servicio, registro de usuarios, enrutamiento, tarifación, etc.

- ✓ **Gateways.-** Su misión es la de enlazar la red VoIP con la red telefónica analógica o RDSI, actuando de forma transparente para el usuario.

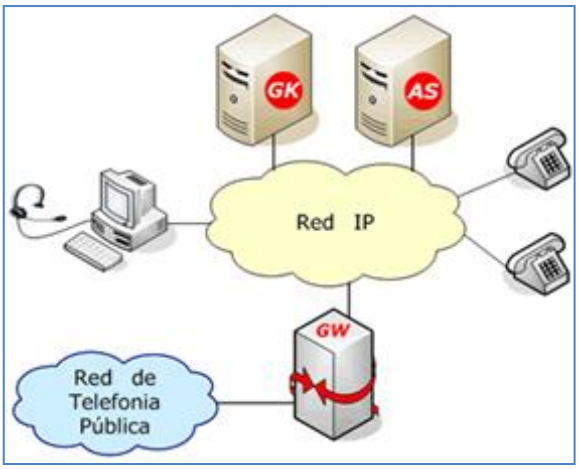

Figura II. 2 Arquitectura de VoIP
Fuente: http://www.teleco.com.br/es/es_tecvoip.asp

Con estos tres elementos, la estructura de la red VoIP podría ser la conexión de dos delegaciones de una misma empresa. La ventaja es inmediata, todas las comunicaciones entre las delegaciones son completamente gratuitas. Este mismo esquema se podría aplicar para proveedores, con el consiguiente ahorro que esto conlleva.

2.1.2. EQUIPAMIENTO PARA VOIP

2.1.2.1. Teléfono IP

Equipo diseñado para conectarse con una red de telefonía IP.

Un teléfono IP no tiene muchas diferencias con respecto a un teléfono analógico, la diferencia más importante es que en el teléfono analógico solamente tenemos un puerto RJ11 y en un teléfono digital se dispone de uno o varios puertos Ethernet (RJ45), un conector para la electricidad, y un puerto RJ11 (opcional) para conectar un headset o similar.

Figura II. 3 Teléfono IP marca CISCO

Algunas marcas de teléfonos IP son:

- ✓ Polycom
- ✓ Cisco
- ✓ Linksys
- ✓ Avaya
- ✓ Snom
- ✓ Aastra
- ✓ Grandstream

2.1.2.2. Softphone

Un softphone (en inglés combinación de software y de telephone) es un software que hace una simulación de teléfono convencional por computadora. Es decir, permite usar

la computadora para hacer llamadas a otros softphones o a otros teléfonos convencionales.

Ejemplos de softphone son:

- ✓ X-lite
- ✓ Zoiper
- ✓ Ekiga
- ✓ Xten EyeBeam
- ✓ Xten Bria Communicator

Figura II. 4 Softphone X-lite versión 4

2.1.2.3. Puertos FXS/FX0

Son Interfaces usadas para la conexión con el exterior mediante circuitos analógicos, es decir que conectan una PBX con la PSTN.

FXS (Foreign eXchange Subscriber).- Permite conectar teléfonos analógicos que interactúan con la central telefónica, este puerto envía señales de timbre y tono.

FXO (Foreign eXchange Office).- Puerto donde se conecta la línea analógica de la PSTN. Este puerto no envía señales de tono o timbrado, solo recibe dichas señales.

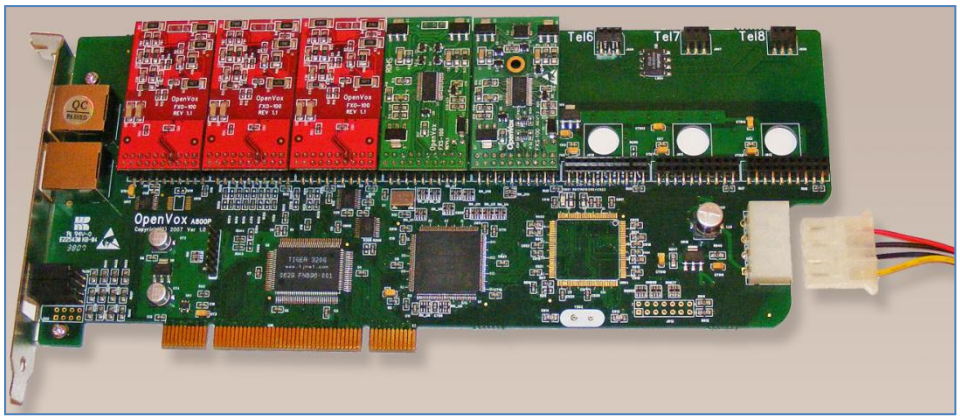

Figura II. 5 Tarjeta analógica con 3 puertos FXO y 2 FXS instalados
Fuente: Comunicaciones Unificadas con Elastix-Volumen1, Edgar Landivar

2.1.2.4. Adaptador de Teléfono Analógico (ATA)

Es un dispositivo usado para conectar uno o más teléfonos analógicos a un sistema de telefonía digital tal como lo es VoIP.

Un ATA por lo general toma la forma de una pequeña caja con un adaptador de corriente, un puerto Ethernet, uno o más puertos FXS y también puede tener un enlace FXO.

El más común ATA cuenta con al menos un puerto FXS, usado para conectar un teléfono convencional, y el conector Ethernet para conectar el adaptador a una red LAN.
Este dispositivo se comunica con el servidor utilizando un protocolo como el H.323, SIP, MGCP, SCCP o IAX, y codifica y decodifica la señal de voz utilizando un codec de voz como G.711, G.729, GSM, iLBC u otros.

Se utiliza aproximadamente de 3 a 5 vatios de electricidad, dependiendo del modelo y marca.

Figura II. 6 ATA marca Grandstream
Fuente: http://linuxman.blogsome.com/2010/09/03/grandstream-handtytone-286-en-elastix/

2.1.3. PROTOCOLOS DE VOIP

Hay muchos protocolos involucrados en la transmisión de Voz sobre IP, en la siguiente figura podemos observar a SIP, RTP, TCP, UDP, entre otros.

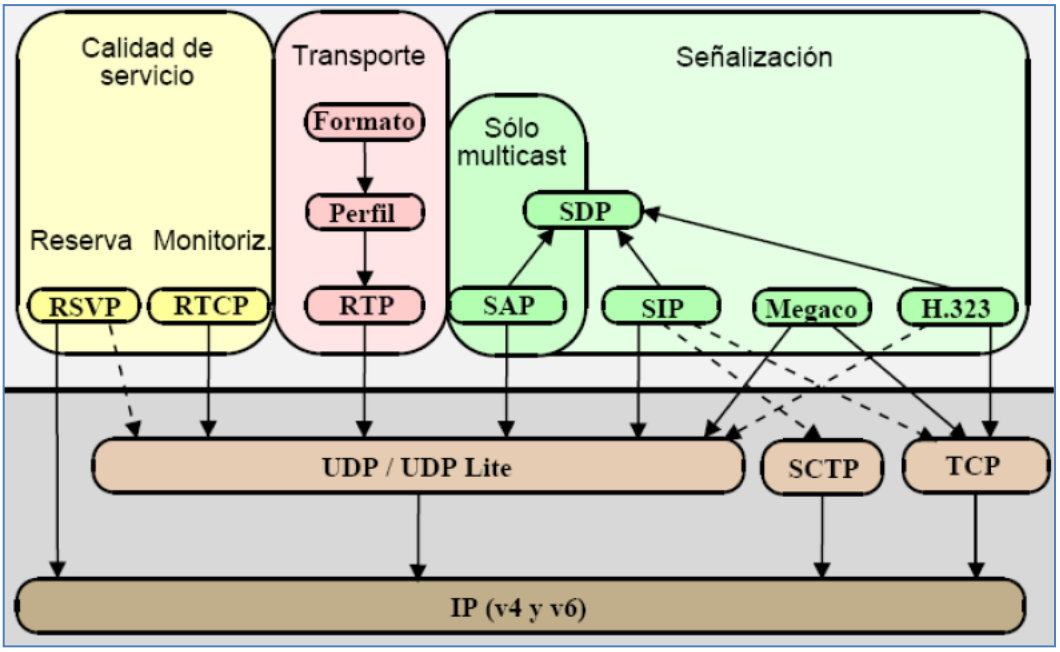

Figura II. 7 Protocolos de VoIP
Fuente: http://www.voip.unam.mx/archivos/docs/Curso%20SIP_05012008.pdf

2.1.3.1. Protocolos de Señalización

Los protocolos de señalización en VoIP cumplen funciones similares a sus homólogos en la telefonía tradicional, es decir tareas de establecimiento de sesión, control del progreso de la llamada, entre otras. Se encuentran en la capa 5 del modelo OSI, es decir en la capa de Sesión.

Existen algunos protocolos de señalización, que han sido desarrollados por diferentes fabricantes u organismos como la ITU o el IETF, y que se encuentran soportados por Asterisk. Algunos son:

SIP.- Protocolo definido por el IETF, posteriormente se hablará más acerca de este protocolo.

IAX (Inter-Asterisk eXchange protocol).- Ahora se refiere generalmente al IAX2, la segunda versión del protocolo IAX, ya que el protocolo original ha quedado obsoleto.

- Es un protocolo abierto.
- Aún no es un estándar.
- Utiliza el puerto UDP 4569 tanto para señalización de canal como para RTP (Protocolo de Transporte en tiempo Real).
- Puede truncar o empaquetar múltiples sesiones dentro de un flujo de datos, así requiere de menos ancho de banda y permite mayor número de canales entre terminales.
- En seguridad, permite la autenticación, pero no hay cifrado entre terminales.
- Según la documentación (Asterisk 1.4) el IAX puede usar cifrado (aes128), siempre sobre canales con autentificación MD5.

H.323.- Utilizado comúnmente para VoIP y para videoconferencia basada en IP. Es un conjunto de normas ITU para comunicaciones multimedia que hacen referencia a los terminales, equipos y servicios estableciendo una señalización en redes IP. No garantiza una calidad de servicio, y en el transporte de datos puede, o no, ser fiable; en el caso de voz o vídeo, nunca es fiable. Además, es independiente de la topología de la red y admite pasarelas, permitiendo usar más de un canal de cada tipo (voz, vídeo, datos) al mismo tiempo.

MGCP (Media Gateway Control Protocol).- Es un protocolo de control de dispositivos, donde un gateway esclavo (MG, Media Gateway) es controlado por un maestro (MGC, Media Gateway Controller, también llamado Call Agent).
Se diferencia del resto de los protocolos VoIP por ser del tipo cliente – servidor. No es un estándar.

SCCP (Skinny Client Control Protocol)

- Es un protocolo propietario de Cisco.
- Es el protocolo por defecto para terminales con el servidor Cisco Call Manager PBX que es el similar a Asterisk PBX.
- El cliente Skinny usa TCP/IP para transmitir y recibir llamadas.
- Para el audio utiliza RTP, UDP e IP.
- Los mensajes Skinny son transmitidos sobre TCP y usa el puerto 2000.

Entre estos, los más populares en el ámbito de Asterisk son SIP e IAX.

2.1.3.2. Protocolos de Transporte de Voz

No se debe confundir con protocolos de transporte de bajo nivel como TCP y UDP. Aquí se habla del protocolo que transporta la voz propiamente dicha o lo que comúnmente se denomina carga útil. Este protocolo se llama RTP.

RTP (Real-time Transport Protocol).- Es de capa aplicación y su función es simple, transportar la voz con el menor retraso posible.
Este protocolo entra a funcionar una vez que el protocolo de señalización ha establecido la llamada entre los participantes.

RTP va de la mano de su protocolo de control, RTCP: RTP envía los datos y RTCP proporciona servicios de control y otras funcionalidades. Existe una variante llamada SRTP (Secure RTP) usada para aportar características de cifrado al canal RTP.

RTCP (RTP Control Protocol).- Se encarga de monitorizar la calidad del servicio y de proporcionar información acerca de los participantes en una sesión de intercambio de datos.

RTSP (Real-Time Streaming Protocol).- Protocolo usado para sistemas de vídeo bajo demanda, optimiza el flujo de datos multimedia. En sintaxis y funcionamiento, es similar al protocolo HTTP, donde tanto el cliente y el servidor pueden hacer peticiones. No obstante, a diferencia de HTTP, el protocolo RTSP necesita mantener información de estado. Además, es independiente de la capa de transporte usada: puede utilizar tanto TCP como UDP.

RSVP (Resource reSerVation Protocol).- Usado para manejar la calidad de servicio de la comunicación. El propósito de RSVP es eliminar aquellas situaciones en las que la voz se pierde porque tenemos una ráfaga de datos en la red. Para ello, éste solicita ancho de banda, divide los paquetes de datos grandes y da prioridad a los paquetes de voz cuando hay una congestión en un router.

2.1.3.3. Protocolos de Plataforma IP

En esta categoría se agrupa a los protocolos básicos en redes IP y que forman la base sobre la cual se añaden los protocolos de voz anteriores. Entre ellos, IP, TCP y UDP.

2.1.4. SIP (SESSION INITIATION PROTOCOL)

El Protocolo de Inicio de Sesión se encarga de iniciar, mantener y terminar sesiones multimedia, las cuales se llevan a cabo de manera interactiva. Por sesiones multimedia se refiere a aplicaciones de audio, video, mensajería instantánea, conferencias y aplicaciones similares.

Debido a que SIP es solo un protocolo de señalización, una vez establecida la sesión, los participantes de la sesión intercambian directamente su tráfico audio/video a través del protocolo RTP.

2.1.4.1. Características

- ✓ **Localización del usuario.-** Posee la capacidad de conocer en todo momento la localización de los usuarios. De esta manera no importa en qué lugar se encuentre un determinado usuario. En definitiva la movilidad de los usuarios no se ve limitada.
- ✓ **Negociación de parámetros.-** Posibilidad de negociar los parámetros necesarios para la comunicación: puertos para el tráfico SIP, direcciones IP, codecs, etc.
- ✓ **Disponibilidad del usuario.-** Permite determinar si un determinado usuario está disponible o no para establecer una comunicación.
- ✓ **Gestión de la comunicación.-** Permite la modificación, transferencia, finalización de la sesión activa. Además informa del estado de la comunicación que se encuentra en progreso.

2.1.4.2. Entidades

SIP cuenta con dos entidades fundamentales, los agentes de usuario y los servidores.

a. **Agentes de Usuario (User Agent-UA).-** A su vez los UA se dividen en:

- ✓ **User Agent Client (UAC).-** Es una entidad lógica que genera peticiones SIP y recibe respuestas a esa peticiones por parte de UAS. Un ejemplo de UAC es un teléfono IP ya que realiza peticiones SIP.

- ✓ **User Agent Server (UAS).-** Es una entidad lógica que recibe peticiones del UAC y genera respuestas a dichas peticiones. Un teléfono IP también es un ejemplo de UAS, ya que acepta las peticiones de inicio de comunicación enviadas por otro teléfono (UAC). Un servidor SIP o proxy, también es un UAS.

Ambos se encuentran en todos los agentes de usuario, así permiten la comunicación entre diferentes agentes de usuario mediante comunicaciones de tipo cliente-servidor.

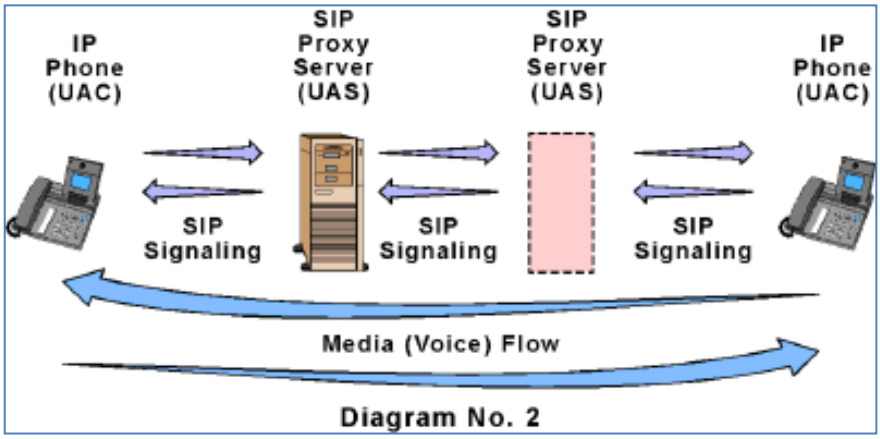

Figura II. 8 Ejemplo de UAC y UAS
Fuente: http://www.mcguireconsulting.comnewsl_sipserver.html

b. Servidores.- Pueden ser de tres tipos:

✓ **Servidor Proxy.-** Recibe pedidos de conexión de un UA y transfiere al UA llamado o a otro servidor proxy si la estación en particular no está en su administración.

Existen dos tipos:

- **Stateful Proxy.-** Mantienen el estado de las transacciones durante el procesamiento de las peticiones y permiten la división de una petición en varias.
- **Stateless Proxy.-** Al contrario de stateful proxy no mantienen estado únicamente se limitan a reenviar los mensajes.

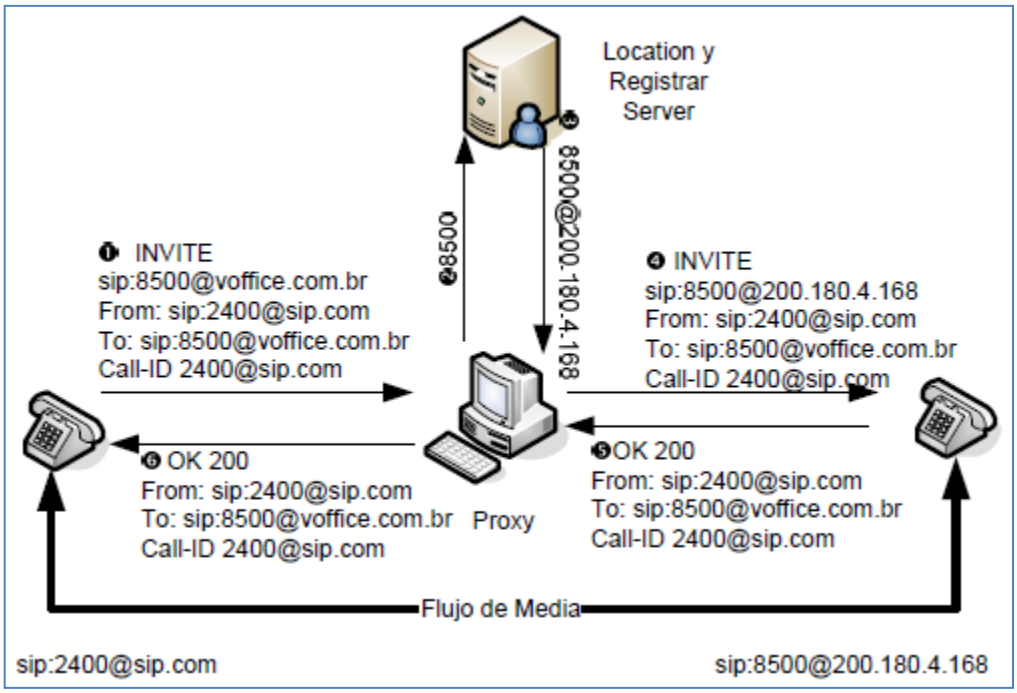

Figura II. 9 Servidor Proxy
Fuente: GONCALVES, F.E. ASTERISK PBX Guía de la Configuración.p 169

- ✓ **Servidor de Registro/Localización.-** Este servidor acepta peticiones de registro de los usuarios y guarda la información de estas para suministrar un servicio de localización y traducción de direcciones en el dominio que controla.

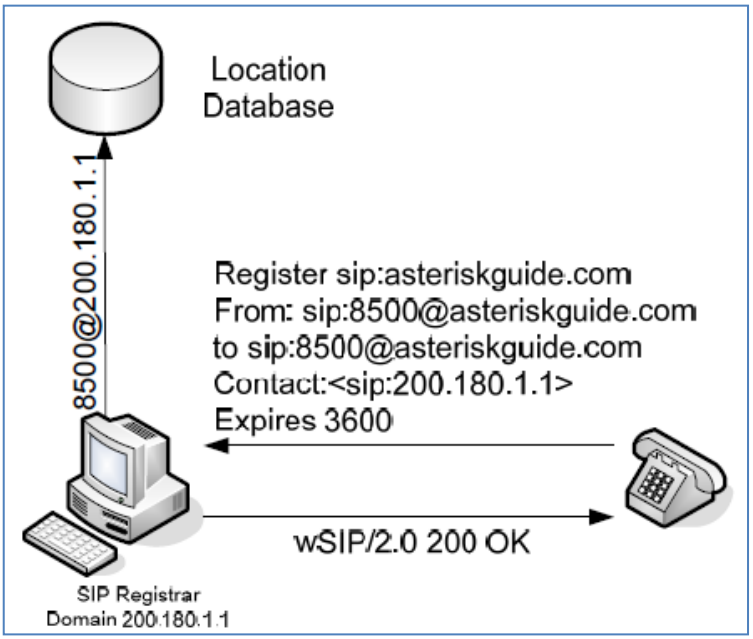

Figura II. 10 Servidor de Registro

Fuente: GONCALVES, F.E. ASTERISK PBX Guía de la Configuración.p 168

- ✓ **Servidor de Redireccionamiento.-** Esta entidad genera respuestas de redirección a las peticiones que recibe y reencamina las peticiones hacia el próximo servidor.

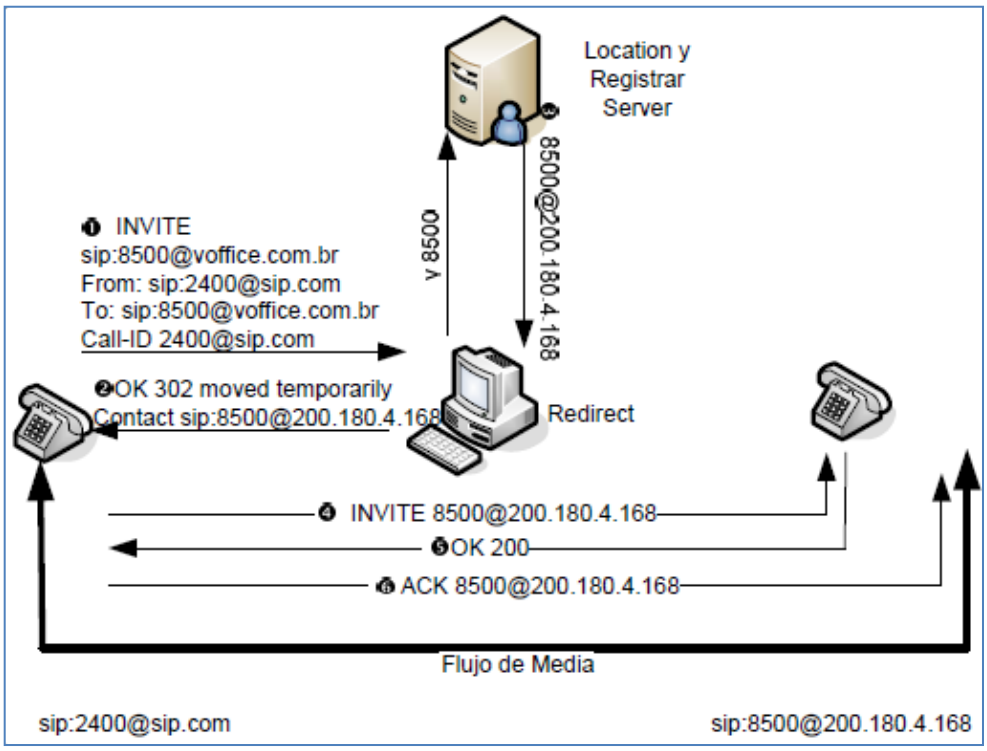

Figura II. 11 Servidor de Redireccionamiento
Fuente: GONCALVES, F.E. ASTERISK PBX Guía de la Configuración.p 169

La división de estos servidores es conceptual, cualquiera de ellos puede estar físicamente en una única máquina, la división de estos puede ser por motivos de escalabilidad y rendimiento.

Back-to-back user agent o B2BUA.- Es una entidad intermediaria que procesa peticiones SIP entrantes comportándose como un UAS, y responde a estas actuando como un UAC regenerando por completo la petición SIP entrante en una nueva petición SIP que va a ser enviada.

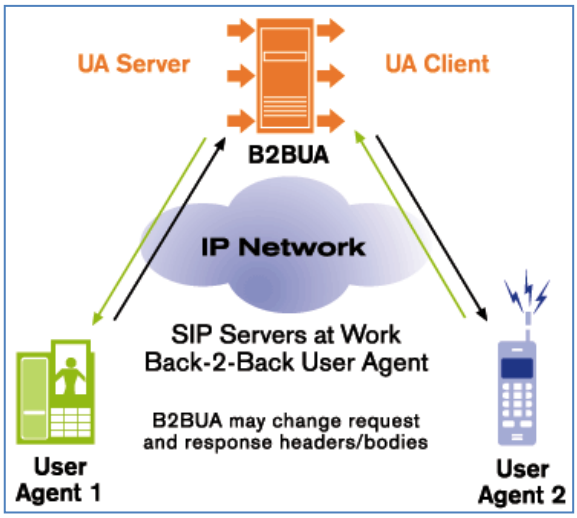

Figura II. 12 Back-to-back user agent o B2BUA
Fuente:http://www.sipcenter.com/sip.nsf/html/Bringing+Telephony+Features+into+SIP+Networks+with+Back+To+Back+User+Agent

2.1.4.3. Direcciones SIP

El protocolo SIP es similar a HTTP por la forma en que funciona (protocolo basado en texto) y es similar a SMTP en la forma en la que se especifican las direcciones SIP.

Las direcciones SIP identifican a un usuario de un determinado dominio. A estas direcciones SIP habitualmente se les llama URI o Uniform Resource Identifier. Una URI se puede especificar de las siguientes maneras:

- sip:usuario@dominio[:port]
- sip:usuario@direcciónIP[:port]

El dominio representa el nombre del proxy SIP que conoce la dirección IP del terminal identificado por usuario de dicho dominio. El puerto por defecto para SIP es 5060, aunque es posible especificar otros adicionales si es necesario.

Tabla II. I Ejemplos de direcciones SIP

DIRECCIÓN SIP	DESCRIPCIÓN
100@voip.com	Usuario 100 perteneciente al dominio voip.com
100@192.168.1.200	Usuario 100 perteneciente al dominio con dirección IP 192.168.1.200

2.1.4.4. Mensajes

Un mensaje se conforma de:

- ✓ **Línea Inicial.-** Puede ser Request-Line o Status-Line.
- ✓ **Encabezado.-** Información relacionada con la llamada (texto plano); por ejemplo: origen y destino de la petición, identificador de la llamada, etc.
- ✓ **Cuerpo del mensaje.-** Carga útil (payload) lleva información (SDP ó ISUP en caso de una troncal hacia la PSTN).

Figura II. 13 Formato del Mensaje SIP
Fuente: http://www.voip.unam.mx/archivos/docs/Curso%20SIP_05012008.pdf

Los mensajes SIP pueden ser de dos tipos, peticiones o respuestas.

a) **Peticiones.-** Las peticiones SIP son caracterizadas por la línea inicial del mensaje, llamada **Request-Line** (Ver Figura 14), que contiene el nombre del método, el identificador del destinatario de la petición (Request-URI) y la versión del protocolo SIP.

Figura II. 14 Línea inicial de una Petición SIP
Fuente: http://dtm.unicauca.edu.co/pregrado/conmutacion/transp/5.2-SIP.pdf

Existen seis métodos básicos SIP que describen las peticiones de los clientes.

Tabla II. II Peticiones SIP

PETICIÓN	FUNCIÓN
INVITE	Permite invitar un usuario o servicio para participar en una sesión o para modificar parámetros en una sesión ya existente.
ACK	Confirma que el extremo llamado recibió el INVITE, es decir confirma el establecimiento de una sesión.
OPTION	Solicita información sobre las capacidades de un servidor. Un UA puede enviar peticiones OPTIONS a un UAS para solicitar cierta información sobre este.
BYE	Para finalizar una sesión.
CANCEL	Para cancelar una sesión que no se ha completado del todo.
REGISTER	Para que el proxy conozca la localización del emisor del mensaje.

b) **Respuestas.-** Las respuestas se generan como retorno de una petición devolviendo un código de estado. En este caso la línea inicial recibe el nombre de **Status-Line** (Ver Figura 15)**,** que llevará la versión del SIP utilizado, código de respuesta y una pequeña descripción de ese código.

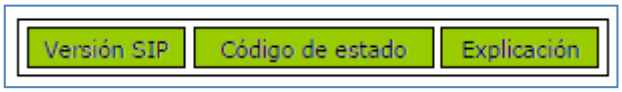

Figura II. 15 Línea inicial de una Respuesta SIP
Fuente: http://dtm.unicauca.edu.co/pregrado/conmutacion/transp/5.2-SIP.pdf

El código de la respuesta está compuesto por tres dígitos que permiten clasificar los diferentes tipos existentes. El primer dígito define la clase de la respuesta.

Tabla II. III Respuestas SIP

CÓDIGO	FUNCIÓN
1XX	Respuestas Provisionales, informativas
2XX	Éxito
3XX	Redirección
4XX	Fallo del cliente, errores de solicitud
5XX	Fallo del servidor
6XX	Fallos globales

2.1.4.5. Transacciones

Una transacción SIP es una secuencia de mensajes entre dos elementos de red. Corresponde a una petición y todas sus respuestas provisionales y una o más respuestas finales (INVITE puede ser dividido por un proxy, por lo tanto tendrá múltiples respuestas finales).

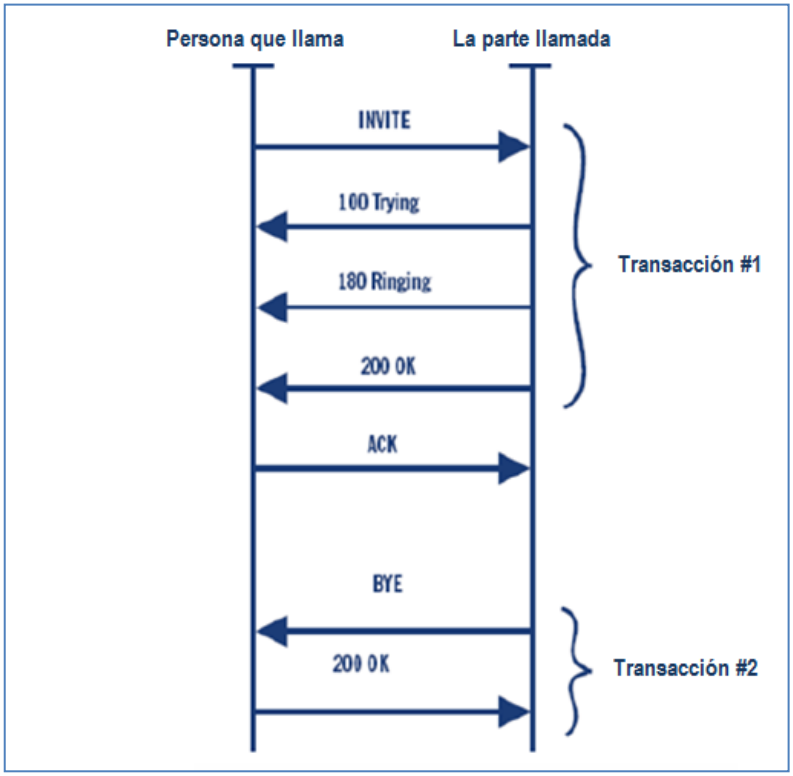

Figura II. 16 Ejemplo de transacción SIP
Fuente: http://www.voip.unam.mx/archivos/docs/Curso%20SIP_05012008.pdf

2.1.5. CODECS

Los Codecs son usados para convertir una señal analógica de voz en una versión codificada digitalmente. Los Codecs varían en calidad de sonido, banda ancha necesaria y requisitos computacionales. Cada servicio, programa, teléfono o gateway, típicamente, soporta varios codecs diferentes y cuando van a hablar uno con otro negocian que codec es el que van a usar.

Además de la ejecución de la conversión de analógico a digital, el CODEC comprime la secuencia de datos, y proporciona la cancelación del eco. La compresión de la forma de onda representada puede permitir el ahorro del ancho de banda. Esto es especialmente interesante en los enlaces de poca capacidad y permite tener un mayor número de conexiones de VoIP simultáneamente. Otra manera de ahorrar ancho de banda es el uso de la supresión del silencio, que es el proceso de no enviar los paquetes de la voz entre silencios en conversaciones humanas.

A continuación se muestra una tabla con los codecs utilizados en VoIP:

Tabla II. IV CODECS utilizados en VoIP

Nombre	Estandarizado	Bit rate (Kbps)	MOS
G.711	ITU-T	56 o 64	4.1
G.722	ITU-T	48, 56 o 64	
G.723.1	ITU-T	5.3 o 6.4	3.8 – 3.9
G.726	ITU-T	16, 24, 32 o 40	3.85
G.728	ITU-T	16	3.61
G.729	ITU-T	8	3.92
GSM 06.10	ETSI	13	
iLBC		8	
LPC10	Gobierno de USA	2.4	
Speex		8, 16 o 32	

Bit rate: indica la cantidad de información que se manda por segundo.
MOS (Mean Opinion Score): indica la calidad general del códec (valor de 1 a 5).

2.2. CALL CENTER - CC (CENTRO DE LLAMADAS)

Un Centro de Atención de Llamadas, o Call Center, es un área donde se recogen las llamadas masivas con destino a una empresa o servicio y que, al mismo tiempo, sirve para realizar llamadas salientes.

Cada vez existen más de ellos y constituyen el principal medio de contacto de las empresas con sus clientes.

2.2.1. CONCEPTO

El Call Center es una solución que se deriva del concepto de la Integración Computador-Teléfono (CTI, Computer Telephone Integration), es decir la interacción física y funcional entre un sistema telefónico y un sistema informático que facilita el intercambio de información.

Teniendo esto en cuenta, podemos definir un Call Center como El conjunto tecnológico y administrativo que permite unificar la inteligencia y potencia de procesamiento de los sistemas informáticos y las facilidades de la conmutación de llamadas telefónicas, para suministrar información a los llamantes en un ambiente de intimidad personal.

2.2.2. SERVICIOS

Entre los servicios ofrecidos por un Call Center figuran:

SECTOR TELECOMUNICACIONES

- Servicio al cliente
- Reporte de Fallas

SERVICIOS DE EMERGENCIA

- Números 900

SECTOR SALUD

- Información
- Programación de Citas

SECTOR FINANCIERO

- Información de Transacciones
- Encuestas a los usuarios

SECTOR TRANSPORTE AEROLINEAS

- Venta de Tiquetes
- Horarios de Vuelo

SECTOR HOTELERO

- Reservaciones

Otros Servicios

- Asesoramiento Técnico
- Telemarketing
- Televenta
- Servicio post-venta.
- Recepción de pedidos, etc.

2.2.3. COMPONENTES DE UN CENTRO DE ATENCIÓN DE LLAMADAS

En un Centro de llamadas se distinguen los siguientes componentes:

- ✓ Agentes
- ✓ PBX
- ✓ ACD
- ✓ IVR
- ✓ Servidor de Base de Datos
- ✓ Servidor CTI

2.2.3.1. AGENTES (Operadores)

Aquellas personas que contestan y/o realizan las llamadas telefónicas. Además de contestar las llamadas, también tienen la capacidad de asesorar y atender cualquier inquietud de los usuarios.

Los agentes pueden ser:

- Agente Inbound: Labores exclusivas de entrada.
- Agente Outbound: Labores exclusivas de salida.
- Agente Blend: Labores de entrada y de salida.

2.2.3.2. PBX

La Central Telefónica es el elemento básico de toda la infraestructura. Su misión es gestionar las extensiones telefónicas corporativas internas y conectarse a la Red Telefónica Pública Conmutada (RTPC-PSTN) para la comunicación con el exterior.

Los clientes de una Central Telefónica IP pueden ser teléfonos IP, softphone o teléfonos analógicos con un adaptador.

2.2.3.3. ACD - Distribuidor Automático de Llamadas

Es el centro de despachos de las llamadas entrantes, este sistema recibe las llamadas de sus clientes con un mensaje personalizado y luego las pone en fila de espera hasta que puedan ser atendidas por un operador. Debido a esta distribución automática se equilibra las cargas de trabajo en los operadores.

Puede ser un sistema autónomo o una capacidad de ACD incorporada en una central telefónica.

Los distribuidores automáticos de llamadas, en general consisten en hardware, tales como servidores, routers, teléfonos, y también software que provee la funcionalidad de flujo de llamada.

2.2.3.4. IVR - Respuesta de Voz Interactiva

Consiste en un sistema telefónico capaz de recibir una llamada e interactuar con el humano a través de grabaciones de voz y el reconocimiento de respuestas simples, como "sí", "no" u otras. Es un sistema automatizado de respuesta interactiva, orientado a entregar y/o capturar información a través del teléfono, permitiendo el acceso a servicios de información u otras operaciones.

Funcionamiento

El usuario realiza una llamada a un número de teléfono, el sistema de audiorespuesta contesta la llamada y le presenta al usuario una serie de acciones a realizar, esto se hace mediante mensajes (menús de opciones) previamente grabados en archivos de audio (Por ejemplo "Presione uno para ventas, dos para administración"). El usuario elige la opción a realizar introduciendo un número en el teclado del teléfono y navega por los diferentes menús hasta encontrar la información solicitada o que el sistema enrute la llamada al destinatario elegido.

El IVR es el elemento al cual se le asignan los trabajos de suministro de información rutinaria, dejando para los agentes la atención especializada y específica de los requerimientos de los llamantes.

2.2.3.5. SERVIDOR CTI

Servidor que une la infraestructura informática corporativa y la telefónica. Cuando la llamada llega al agente en la pantalla de éste aparece toda la información del cliente, así, se libera al operador o agente de tareas repetitivas (identificación del cliente).

En el siguiente gráfico se puede observar como el Servidor CTI cumple la función de coordinar todos los componentes hardware y software del CC.

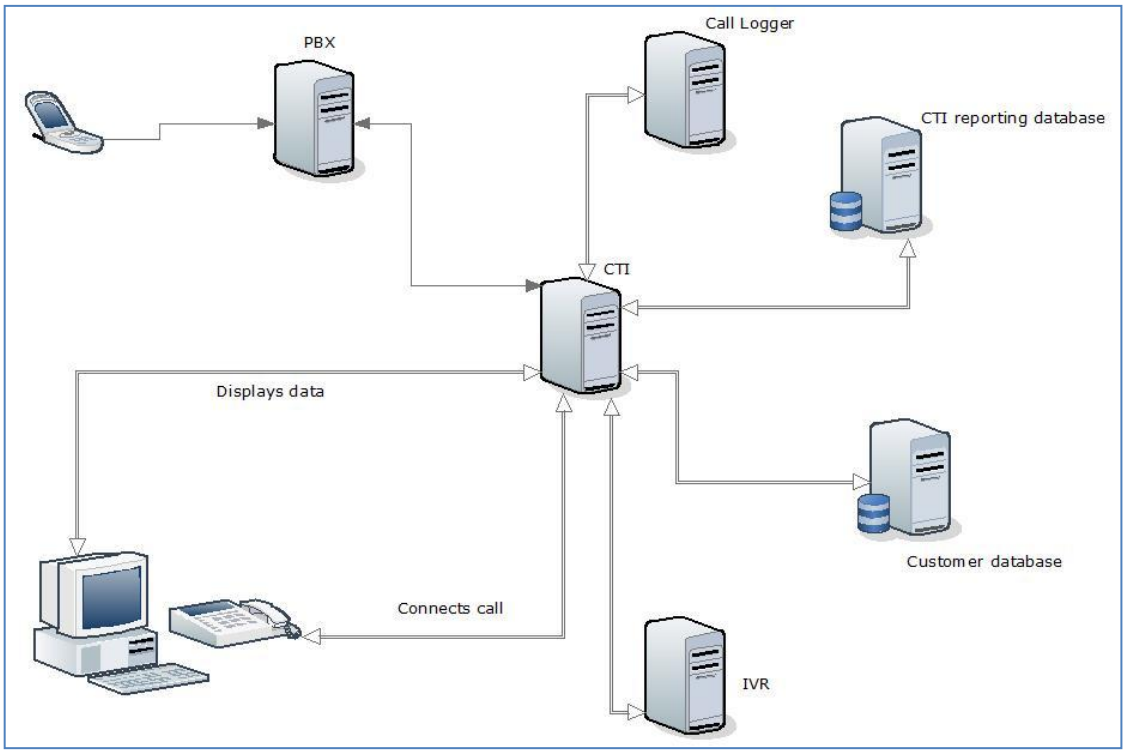

Figura II. 17 Sistema de Integración Telefonía-Computador
Fuente: http://www.siebeloracle.com/siebel-cti-computer-telephony-integration/

CAPÍTULO III

CLUSTERING Y ALTA DISPONIBILIDAD

3.1. CLUSTERS

Un clúster es un grupo de computadores, interconectadas mediante una red, que trabajan conjuntamente y que se comportan como si fuesen un único sistema.

Los clúster son usualmente empleados para mejorar el rendimiento y/o la disponibilidad a niveles que un único sistema no puede alcanzar, o para ser una alternativa económica y equiparable a costosos sistemas de alta potencia y disponibilidad.

La construcción de las computadores del clúster es más fácil y económica debido a su flexibilidad: pueden tener todos la misma configuración de hardware y sistema operativo (clúster homogéneo), diferente hardware pero con arquitecturas y sistemas operativos similares (clúster semi-homogéneo), o tener diferente hardware y sistema operativo (clúster heterogéneo).

Para que un clúster funcione como tal, no basta solo con conectar entre sí los ordenadores, sino que es necesario proveer un sistema de manejo del clúster, el cual se encargue de interactuar con el usuario y los procesos que corren en él para optimizar el funcionamiento.

3.1.1. TIPOS DE CLÚSTER

3.1.1.1. Clúster de Alta disponibilidad (HA, High Availability)

Los clúster de alta disponibilidad tienen como propósito principal brindar la máxima disponibilidad de los servicios que ofrecen. Esto se consigue mediante software que monitoriza constantemente el clúster, detecta fallos y permite recuperarse frente a los mismos.

3.1.1.2. Clúster de Alto Rendimiento (HP, High Performance)

Este tipo de clúster se utiliza para ejecutar programas paralelizables que requieren de gran capacidad computacional de forma intensiva. Son de especial interés para la comunidad científica o industrias que tengan que resolver complejos problemas o simulaciones.

3.1.1.3. Clúster de Balanceo de Carga (LB, Load Balancing)

Permite distribuir las peticiones de servicio entrantes hacia un conjunto de equipos que las procesa. Se utiliza principalmente para servicios de red sin estado, como un servidor web o un servidor de correo electrónico, con altas cargas de trabajo y de tráfico de red.

Las características más destacadas de este tipo de clúster son su robustez y su alto grado de escalabilidad.

3.1.2. COMPONENTES DE UN CLÚSTER

Un sistema clúster está formado habitualmente por diversos componentes hardware y software:

3.1.2.1. Nodos

Cada una de las máquinas que componen el clúster, pueden ser desde simples ordenadores personales a servidores dedicados, conectados por una red. Por regla general los nodos deben tener características similares: arquitectura, componentes, sistema operativo.

3.1.2.2. Sistemas Operativos

Se utilizan sistemas operativos de tipos servidor con características de multiproceso y multiusuario, así como capacidad para abstracción de dispositivos y trabajo con interfaces IP virtuales.

3.1.2.3. Middleware de Clúster

Es el software que actúa entre el sistema operativo y los servicios o aplicaciones finales. Es la parte fundamental del clúster donde se encuentra la lógica del mismo.

3.1.2.4. Sistema de almacenamiento

El almacenamiento puede ir desde sistemas comunes de almacenamiento interno del servidor a redes de almacenamiento compartido NAS o SAN.

Tecnologías en el soporte del almacenamiento en discos duros:

- IDE o ATA: velocidades de 33, 66, 100, 133 y 166 MB/s
- SATA: velocidades de 150, 300 y 600 MB/s
- SCSI: velocidades de 160, 320, 640 MB/s. Proporciona altos rendimientos.
- SAS: aúna SATA-II y SCSI. Velocidades de 300 y 600 MB/s
- Las unidades de cinta (DLT) son utilizadas para copias de seguridad por su bajo coste.

NAS (Network Attached Storage) es un dispositivo específico dedicado al almacenamiento a través de red (normalmente TCP/IP) que hace uso de un sistema operativo optimizado para dar acceso a través de protocolos CIFS, NFS, FTP o TFTP.

Por su parte, DAS (Direct Attached Storage) consiste en conectar unidades externas de almacenamiento SCSI o a una SAN (Storage Area Network) a través de un canal de fibra. Estas conexiones son dedicadas.

Mientras NAS permite compartir el almacenamiento, utilizar la red, y tiene una gestión más sencilla, DAS proporciona mayor rendimiento y mayor fiabilidad al no compartir el recurso.

3.1.2.5. Conexiones de red

Los nodos del clúster pueden conectarse mediante una simple red Fast Ethernet o utilizar tecnologías de red avanzadas como Gigabit Ethernet, Infiniband, Myrinet, SCI, etc.

3.1.2.6. Protocolos de Comunicación

Definen la intercomunicación entre los nodos del clúster.

3.1.2.7. Servicios y Aplicaciones

Son aquellos servicios y aplicaciones a ejecutar sobre el clúster.

3.2. ALTA DISPONIBILIDAD (HIGH AVAILAVILITY-HA)

En la actualidad las organizaciones dependen cada vez más de sus sistemas de información, y como es obvio se desea que estos sean seguros y permanezcan disponibles el mayor tiempo posible.

Para una empresa, una interrupción del sistema supone un problema por las consecuencias que tiene en su negocio. Pero es aun más importante la disponibilidad cuando se habla de servicios de misión crítica, donde se debe proporcionar un servicio ininterrumpido de 24 horas al día, 7 días a la semana.

La **Alta Disponibilidad** es la característica que tiene un sistema para protegerse o recuperarse de interrupciones o caídas, de forma automática y en un corto plazo de tiempo.

Los sistemas de alta disponibilidad se diseñan para eliminar o tolerar los posibles puntos de fallo, para ello se emplea principalmente la redundancia interna de componentes (red, almacenamiento, fuentes de alimentación, etc.) así como de los elementos de infraestructura (sistema eléctrico, electrónica de red, etc).

3.2.1. Tiempo de Inactividad

Si un usuario no puede acceder a un sistema se dice que está no disponible. El término tiempo de inactividad (downtime) es usado para definir cuándo el sistema no está disponible.

Existen dos diferentes tiempos de inactividad:

- ✓ **Tiempo de Inactividad Planificado.-** Cuando se paraliza el sistema para realizar cambios o mejoras en hardware o software.

- ✓ **Tiempo de Inactividad no Planificado.-** Hace referencia a los tiempos que surgen por acontecimientos imprevistos como un apagón, un error de hardware o de software, problemas de seguridad, un desastre natural, virus, accidentes, caídas involuntarias del sistema.

3.2.2. Niveles de Disponibilidad

La disponibilidad es usualmente expresada como un porcentaje del tiempo de funcionamiento en un año, que resulta dividiendo el tiempo durante el cual el servicio está disponible por el tiempo total de minutos en una año (525600).

Valores comunes de disponibilidad, típicamente enunciado como número de "nueves" para sistemas altamente disponibles se muestran en la siguiente tabla:

Tabla III. V Porcentajes de Disponibilidad

Porcentaje de Disponibilidad	Tiempo de Inactividad (minutos/mes)	Tiempo de Inactividad (minutos/año)
99,9%	43.8	8.76 (horas/año)
99,99%	4.38	52.6
99,999%	0.44	5.26

Se puede apreciar que estos valores de disponibilidad son visibles mayormente en documentos de ventas o marketing, en lugar de ser una especificación técnica completamente medible y cuantificable.

3.3. CLÚSTER DE ALTA DISPONIBILIDAD

Para conseguir redundancia y protección contra fallos de un sistema, la primera de las medidas que se suelen tomar es replicar sus componentes hardware más críticos. Por ejemplo en el caso de un servidor se emplean configuraciones de discos en RAID, fuentes de alimentación redundantes, varias interfaces de red en bonding, etc. Y el mismo concepto de redundancia se aplica también para el resto de componentes como la electrónica de red o el sistema eléctrico.

Estas medidas indudablemente aumentan el nivel de disponibilidad de un sistema, pero para conseguir un nivel aun más alto, se suelen utilizar configuraciones avanzadas de hardware y software como son los Clúster de Alta Disponibilidad.

Un **Clúster de Alta Disponibilidad** es un conjunto de dos o más servidores, que se caracteriza por compartir recursos y porque están constantemente monitorizándose entre sí.

Si se produce un fallo de hardware o de los servicios de alguna de las maquinas que forman el clúster, el software de alta disponibilidad es capaz de rearrancar

automáticamente los servicios que han fallado en cualquiera de los otros equipos del clúster. Cuando el servidor que falló se recupere, los servicios pueden o no migrar de regreso hacia el nodo original, dependiendo de la configuración establecida por el administrador.

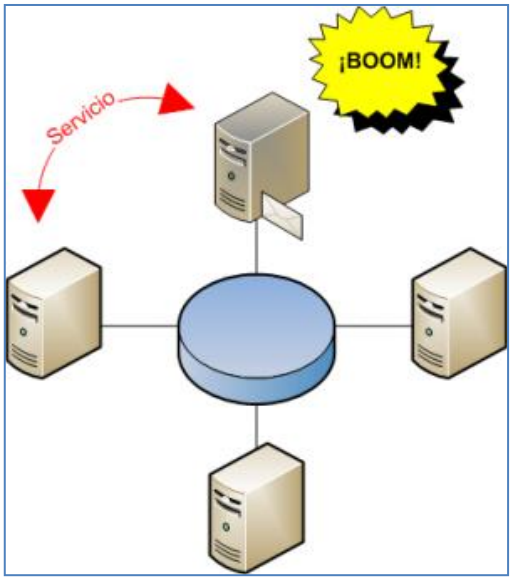

Figura III. 18 Clúster de Alta Disponibilidad
Fuente:http://www.adminso.es/index.php/Aspectos_b%C3%A1sicos_de_alta_disponibilidad_y_alto_rendimiento

Esta capacidad de los clusters de restablecer en pocos segundos un servicio, manteniendo la integridad de los datos, permite que en muchos casos los usuarios no tengan por que notar que se ha producido un problema. Cuando una avería de este tipo, en un sistema sin clúster, podría dejarles sin servicio durante horas.

La utilización de clusters no solo es beneficiosa para caídas de servicio no programadas, sino que también es útil en paradas de sistema programadas como puede ser un mantenimiento hardware o una actualización software.

En general las razones para implementar un clúster de alta disponibilidad son:

- ✓ Aumentar la disponibilidad
- ✓ Mejorar el rendimiento
- ✓ Escalabilidad

- ✓ Tolerancia a fallos
- ✓ Recuperación ante fallos en tiempo aceptable
- ✓ Reducir costes
- ✓ Consolidar Servidores
- ✓ Consolidar el Almacenamiento

3.3.1. CONFIGURACIONES DE ALTA DISPONIBILIDAD

Las configuraciones más comunes en entornos de clúster de alta disponibilidad son la configuración activo/activo y la configuración activo/pasivo.

3.3.1.1. Configuración Activo/Activo

En una configuración activo/activo, todos los servidores del clúster pueden ejecutar los mismos recursos simultáneamente. Es decir, los servidores poseen los mismos recursos y pueden acceder a estos independientemente de los otros servidores del clúster. Si un nodo del sistema falla y deja de estar disponible, sus recursos siguen estando accesibles a través de los otros servidores del clúster.

La ventaja principal de esta configuración es que los servidores en el clúster son mas eficientes ya que pueden trabajar todos a la vez. Sin embargo, cuando uno de los servidores deja de estar accesible, su carga de trabajo pasa a los nodos restantes, lo que produce una degradación del nivel global de servicio ofrecido a los usuarios.

En la siguiente figura se muestra como ambos servidores están activos, proporcionando un mismo servicio a los diferentes usuarios. Los clientes acceden al servicio o recursos de forma transparente y no tienen conocimiento de la existencia de varios servidores formando un clúster.

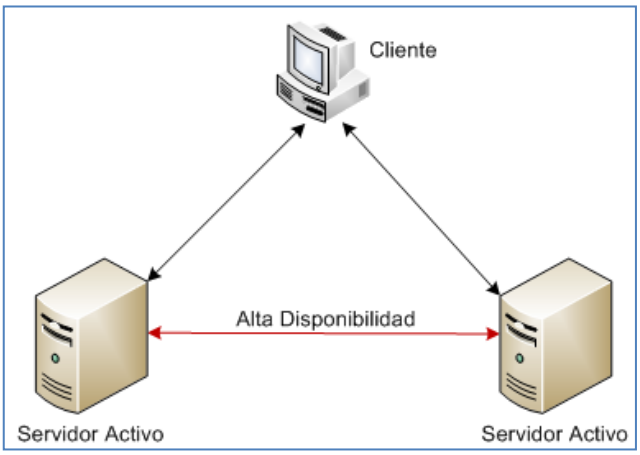

Figura III. 19 HA Activo-Activo
Fuente: http://www.lintips.com/?q=node/119

3.3.1.2. Configuración Activo/Pasivo

En este tipo de cluster solamente hay un nodo que da servicio, es decir que hay un servidor que posee los recursos del cluster y otros servidores que son capaces de acceder a esos recursos, pero no los activan hasta que el propietario de los recursos ya no esté disponible.

Las ventajas de la configuración activo/pasivo son que no hay degradación de servicio y que los servicios solo se reinician cuando el servidor activo deja de responder. Sin embargo, una desventaja de esta configuración es que los servidores pasivos no proporcionan ningún tipo de recurso mientras están en espera, haciendo que la solución sea menos eficiente que el cluster de tipo activo/activo. Otra desventaja es que los sistemas tardan un tiempo en migrar los recursos (failover) al nodo en espera.

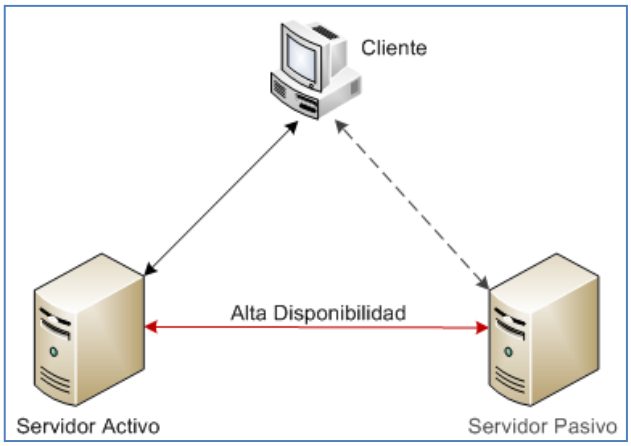

Figura III. 20 HA Activo-Pasivo
Fuente: http://www.lintips.com/?q=node/119

3.3.2. CONCEPTOS BÁSICOS

En un clúster de alta disponibilidad, el software de clúster realiza dos funciones fundamentales. Por un lado intercomunica entre sí todos los nodos, monitorizando continuamente su estado y detectando fallos. Y por otro lado administra los servicios ofrecidos por el clúster, teniendo la capacidad de migrar dichos servicios entre diferentes servidores físicos como respuesta a un fallo.

A continuación se describen los conceptos básicos en el funcionamiento del clúster.

3.3.2.1. Intercomunicación

El software de clúster gestiona servicios y recursos en los nodos. Pero además, tiene que mantener continuamente entre estos una visión global de la configuración y estado del clúster. De esta forma, ante el fallo de un nodo, el resto conoce que servicios se deben restablecer.

Ya que la comunicación entre los nodos del clúster es crucial para el funcionamiento de este, es habitual utilizar un canal específico como una red IP independiente o una conexión serie, que no se pueda ver afectada por problemas de seguridad o rendimiento.

3.3.2.2. Recurso y Grupos de Recursos

Tradicionalmente se entiende como servicio a un conjunto de procesos que se ejecutan en un momento dado sobre un servidor y sistema operativo. Este último provee a los procesos de los recursos necesarios para realizar su tarea: sistema de ficheros, interfaces de red, tiempo de cpu, memoria, etc.

En un clúster de alta disponibilidad, el software de clúster, abstrae e independiza a los servicios de un host concreto. Posibilitando que estos se desplacen entre diferentes servidores de forma trasparente para la aplicación o los usuarios.

El software de clúster permite definir grupos de recursos, que son todos aquellos recursos necesarios por el servicio. Estos recursos serán los scripts de arranque del servicio, un sistema de ficheros, una dirección IP, etc.

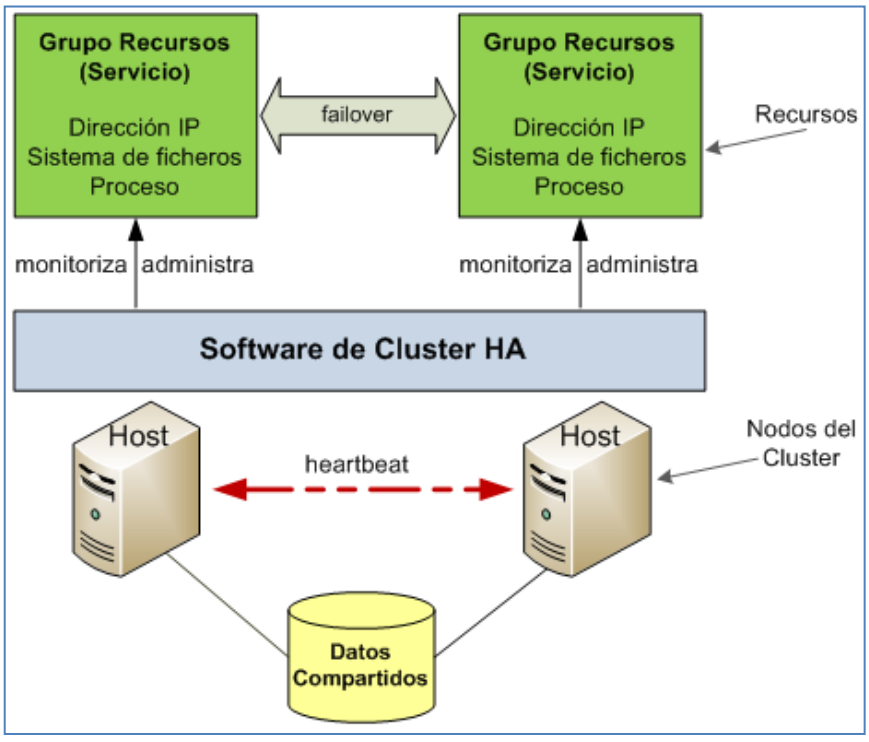

Figura III. 21 Grupo de Recursos de un Cluster
Fuente: http://www.lintips.com/?q=node/119

3.3.2.3. Failover - Migración de Recursos

Cuando un nodo ya no está disponible, o cuando un recurso fallido no se puede reiniciar satisfactoriamente en un nodo, el software de alta disponibilidad reacciona migrando el recurso o grupo de recursos a otro nodo disponible en el clúster.

De este modo el tiempo de inactividad por el posible fallo es mínimo, y el clúster seguirá proporcionando el correspondiente servicio.

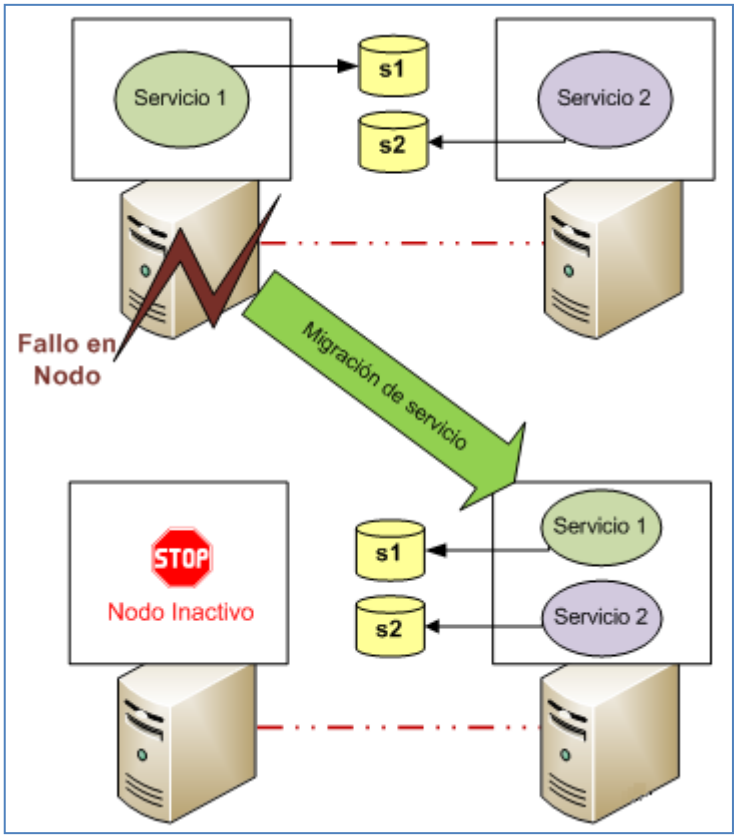

Figura III. 22 Failover
Fuente: http://www.lintips.com/?q=node/119

3.3.2.4. Fencing

En los clusters HA existe una situación donde un nodo deja de funcionar correctamente pero todavía sigue levantado, accediendo a ciertos recursos y respondiendo peticiones. Para evitar que el nodo corrompa recursos o responda con peticiones, los clusters lo solucionan utilizando una técnica llamada Fencing.

La función principal del Fencing es hacerle saber a dicho nodo que está funcionando en mal estado, retirarle sus recursos asignados para que los atiendan otros nodos, y dejarlo en un estado inactivo.

3.3.2.5. Split-Brain

Escenario en el que los nodos del clúster se dividen en dos o más grupos que no saben el uno del otro (ya sea a través de un fallo de software o hardware), provocando que más de un servidor o aplicación pertenecientes a un mismo clúster intenten acceder a los mismos recursos, lo que puede causar daños a dichos recursos.

Este escenario ocurre cuando cada servidor en el clúster cree que los otros servidores han fallado e intenta activar y utilizar dichos recursos.

3.3.2.6. Quorum

Es el nombre que se le da al mecanismo cuyo objetivo es ayudar a resolver el problema SplitBrain. La solución que plantea quorum es la de no seleccionar más de una "partición del clúster", cuando falla la comunicación, es decir, que sólo una partición del clúster ofrezca los servicios.

3.4. SOLUCIONES OPEN SOURCE DE CLUSTERING HA

Existen muchos proyectos Open Source dedicados a proporcionar soluciones para Clusters de Alta Disponibilidad en Linux.

Hay que tener en cuenta también que actualmente las aplicaciones de clustering son bastante complejas por lo que suelen constar de varios componentes. Así que, siguiendo la propia filosofía del software libre, una solución completa de clustering utiliza componentes de varios subproyectos.

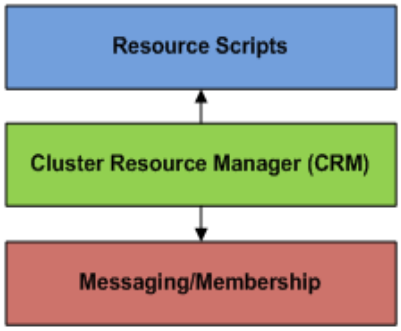

Figura III. 23 Componentes de una Solución Completa de Clustering
Fuente: http://www.lintips.com/?q=node/120

A continuación se indican algunos de los proyectos y componentes más importantes en la actualidad dentro en el ámbito de clusters de Software Libre.

3.4.1. Proyecto Linux-HA

El proyecto Linux-HA tiene como objetivo proporcionar una solución de alta disponibilidad (clustering) para Linux que promueva la fiabilidad y disponibilidad de sistemas.

Linux-HA se utiliza ampliamente y como una parte muy importante, en muchas soluciones de Alta Disponibilidad. Desde que comenzó en el año 1999 a la actualidad, sigue siendo una de las mejores soluciones de software HA para muchas plataformas.
El componente principal de Linux-HA es Heartbeat, del cual se habla posteriormente.

3.4.2. OpenAIS

OpenAIS es una implementación desarrollada por SA Forum que se basa en AIS (Application Interface Specification), un conjunto de especificaciones para estandarizar el desarrollo de servicios e interfaces para la alta disponibilidad.

El proyecto OpenAIS implementa actualmente los componentes de infraestructura y membresía. Y es utilizado en soluciones completas de clustering como Pacemaker o Red Hat Cluster.

3.4.3. Proyecto Red Hat Clúster

Es un proyecto de desarrollo open source de diferentes componentes de clustering para Linux. Esta promovido principalmente por RedHat.
Proporciona además de alta disponibilidad balanceo de carga.

3.4.4. Corosync Cluster Engine

Corosync Cluster Engine es un proyecto open source bajo la licencia BSD, derivado del proyecto Open-AIS. El objetivo principal del proyecto es desarrollar una solución de clúster completa, certificada por la OSI (Open Source Initiative), con soporte para Linux, Solaris, BSD y MacOSX.

El proyecto se inicia en Julio de 2008, y la primera versión estable Corosync 1.0.0 se lanzó en agosto de 2009. Hasta el momento solo la distribución Fedora lo soporta oficialmente incluyendo los binarios de Corosyn en sus repositorios de paquetes.

Otros

Existen otros muchos proyectos dedicados a facilitar la instalación y configuración de clusters de alta disponibilidad. Por ejemplo el proyecto UltraMonkey, que combina LVS + Heartbeat + Ldirector, para proporcionar una solución de clúster HA y balanceo de carga.

Así como otros proyectos de clusters de alta disponibilidad completos que han quedado discontinuados con los años, como el caso de Kimberlite o de OpenHA.

También hay varios proyectos muy interesantes para plataformas diferentes a Linux, como el caso del Open High Availability Cluster (OHAC) que es la versión OpenSource del Solaris Cluster de Sun Microsystems.

3.5. ALTA DISPONIBILIDAD DE DATOS

Existen herramientas cuyo objetivo es mantener los datos replicados sin que presenten problemas de integridad ni consistencia.
La alta disponibilidad de datos se puede clasificar en las siguientes categorías: bloques de datos, bases de datos en MySQL o servidores NAS.

- ✓ **Bloque de Datos.-** Consiste en replicar la información bloque a bloque del sistema de ficheros. Las herramientas mas utilizadas son DRBD, rsync y slony-l.

- ✓ **Bases de Datos.-** Permite replicar una base de datos entre varios servidores. Existen herramientas específicas para cada tipo deservidor (p.e. MySQL, Oracle).

- ✓ **Servidor NAS.-** Permite un nivel de replicación de datos superior ya que crea una unidad de red en la que se guardan todos los datos. En esta categoría las herramientas más usadas son: FreeNAS y Openfiler.

3.6. DISEÑO DE LA SOLUCION DE HA

Para la implementación del prototipo se utilizará Heartbeat en la capa de mensajería y Pacemaker como administrador de los recursos del clúster (CRM). En cuanto a la sincronización de los datos el paquete encargado será Rsync.

A continuación se presente una explicación de cada uno de los paquetes involucrados en el ejemplar de clúster de alta disponibilidad.

3.6.1. HEARTBEAT

Heartbeat es un demonio que proporciona infraestructura de clúster (comunicación y membrecía). Este permite conocer la presencia o caída de un recurso o máquina y facilitar la comunicación entre los nodos.

Para formar una solución clúster de utilidad, Heartbeat necesita combinarse con un CRM, que realiza las tareas de iniciar o parar los recursos (direcciones IP, servicios, etc) a dotar de alta disponibilidad.

Arquitectura de Heartbeat

Históricamente con el nombre Heartbeat se hacía referencia a un conjunto de herramientas y utilidades de alta disponibilidad con una arquitectura estructurada en capas, formado fundamentalmente por el componente Heartbeat en la capa de mensajes, un administrador de recursos locales (LRM), un administrador de recursos del clúster (CRM) en la capa de asignación de recursos, y un agente de recursos (Resource Agent) en la capa de recursos, entre otros componentes.

A partir de la versión 2.1.4 esta generalización ya no es válida, quedando el nombre Hearbeat para denominar exclusivamente la capa de comunicación o mensajes entre los nodos que forman el clúster. El resto de componentes son ahora proyectos independientes, igualmente necesarios para establecer un clúster de alta disponibilidad.

La arquitectura presentada en la actualidad por Linux-HA dispone de una gran flexibilidad debido a la modularidad comentada, algo muy positivo de cara al desarrollo que experimente el proyecto en los próximos años.

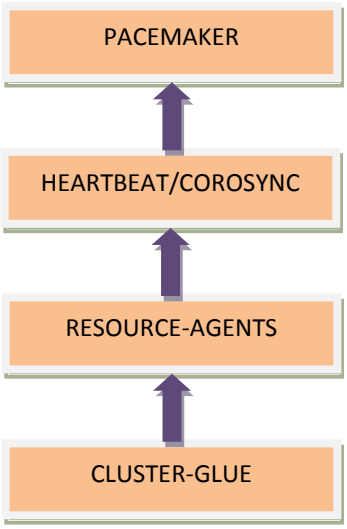

Figura III. 24 Arquitectura del Proyecto Linux-HA

3.6.2. CLÚSTER GLUE

Es un conjunto de bibliotecas, herramientas y utilidades adecuadas para la pila de clúster Heartbeat/Pacemaker. En esencia, Cluster-Glue es todo lo que no es la capa de mensajería del clúster (Heartbeat), ni el administrador de recursos de clúster (Pacemaker), ni un agente de recursos.

Cluster-Glue ha sido manejado como un sub-proyecto de Linux-HA desde su versión 1.0, que coincidió con el lanzamiento de Heartbeat 2.99.

Componentes

a) LRM (Local Resource Manager)

El administrador de recursos locales es la interfaz entre el administrador de recursos del clúster y los agentes de los recursos. Por sí mismo, no aplica ninguna política, simplemente procesa las órdenes recibidas desde el CRM, los pasa a los agentes de recursos, e informa al éxito o el fracaso. En particular, el LRM puede:

- iniciar un recurso
- detener un recurso
- monitorear un recurso
- informar el estado de un recurso

- listar todas las instancias de los recursos que controla en la actualidad, y su estado.

b) STONITH (Shoot The Other Node In The Head)

Se traduciría como Dispara al otro nodo en la cabeza, es un mecanismo para nodo fencing que consiste en apagar o reiniciar a un nodo para asegurarse que libera los recursos que tiene asignados.

c) hb_report

Una utilidad avanzada para el reporte de errores. hb_report genera tarballs que son frecuentemente solicitadas por los desarrolladores para aislar y corregir errores.

d) Cluster Plumbing Library

Biblioteca de bajo nivel para las comunicaciones intra-cluster.

3.6.3. RESOURCE AGENTS (RA)

Un agente de recurso es una interfaz estandarizada para un recurso del clúster. Traduce un conjunto estándar de operaciones en pasos específicos para el recurso o aplicación, e interpreta los resultados como éxito o fracaso.

Al igual que Cluster-Glue, es un subproyecto de Linux-HA desde el lanzamiento de Heartbeat 2.99.

Algunas de las operaciones que un agente de recursos realiza sobre una instancia de recurso son las siguientes:

Tabla III. VI Operaciones de un Agente de Recursos

OPERACIÓN	FUNCIÓN
start	habilitar o iniciar el recurso dado.
stop	deshabilitar o detener el recurso dado.
monitor	comprobar si el recurso dado se está ejecutando y retornar el estado de si está corriendo o no.
valídate-all	validar la configuración del recurso.
meta-data	retornar información sobre el agente de recurso en sí mismo (utilizado por GUIs y otras herramientas de administración y documentación).

Existen tres tipos de agentes de recursos, los mismos que son soportados por Pacemaker:

- **LSB (Linux Standard Base)**

  Los RA LSB se encuentran en */etc/init.d* y generalmente son proporcionados por el sistema operativo.

  Cabe mencionar que parámetros y opciones no pueden ser pasados a agentes de recursos LSB.

- **OCF (Open Cluster Framework)**

  La especificación OCF es básicamente una extensión de LSB, se encuentran en /usr/lib/ocf/resource.d/provider.

  OCF si acepta parámetros en su configuración.

- **Legacy**

  La versión 1 de Heartbeat viene con su propio estilo de agentes de recursos y es muy probable que las personas que desean crear sus propios agentes, los usen como base. Es por ello que el CRM permite que se los siga utilizando.

3.6.4. PACEMAKER

El proyecto Pacemaker surge en el año 2007, a raíz de la segunda generación de Linux-HA. Los programadores del componente CRM de Linux-HA, deciden extraer el desarrollo y mantenimiento de este en un proyecto separado. Para que el nuevo CRM pueda utilizar como capa de comunicación no solo Heartbeat si no también Corosync.

Pacemaker está disponible en la mayoría de las distribuciones Linux actuales, las cuales lo han adoptado como sucesor de Heartbeat. También es la opción elegida en Suse Linux Enterprise, una de las distribuciones comerciales más importantes en el sector corporativo.

3.6.4.1. Características

Pacemaker es un administrador de recursos del clúster, orientado a obtener la máxima disponibilidad de los servicios del clúster (recursos) mediante la detección y recuperación de fallos, haciendo uso de las capacidades de mensajería proporcionada por Heartbeat.

Entre las principales características se destacan:

- ✓ Detección y recuperación de fallos a nivel de nodo y a nivel de recurso.
- ✓ Independencia de almacenamiento, sin necesidad de almacenamiento compartido.
- ✓ Independencia de recursos.
- ✓ Soporta grandes y pequeños clusters.
- ✓ Soporta cualquier configuración de redundancia (Activo/Activo, Activo/Pasivo).
- ✓ Opcionalmente asegura integridad de datos con STONITH.
- ✓ Soporte para servicios los cuales necesitan estar activos en múltiples nodos.
- ✓ La actualización de la configuración se replica automáticamente desde cualquier nodo.

3.6.4.2. Fichero *cib.xml*

Con Pacemaker se gestiona toda la funcionalidad del clúster: nodos, recursos, ficheros de configuración, propiedades globales del clúster, agentes de recursos, etc.
Todos estos aspectos quedan recogidos en el fichero *cib.xml* ubicado en */var/lib/heartbeat/crm/,* el mismo que es posible editarlo con comandos crm.

Este fichero contiene dos secciones:

- sección estática **<configuration>**
- sección dinámica **<status>**

La sección **<configuration>** contiene la configuración del clúster, es decir, las opciones que afectan al funcionamiento del clúster como un todo, los atributos de cada uno de los nodos, las instancias de configuración de los recursos, y las restricciones entre los recursos.

Por otro lado, la sección **<status>** es generada y actualizada de manera dinámica por el CRM, mostrando información sobre el estado actual del clúster: qué nodos se encuentran online y cuáles offline, en qué nodos se encuentran ejecutándose los recursos, cuáles fueron los últimos fallos registrados por Heartbeat, etc.

La estructura del archivo *cib.xml* presenta el siguiente aspecto:

```
<cib>
   <configuration>
          <crm_config/>
          <nodes/>
          <resources/>
          <constraints/>
   </configuration>
   <status/>
</cib>
```

Como se puede observar en el código anterior la sección de configuración contiene a su vez otras cuatro sub-secciones.

a) Opciones de Configuración del clúster

b) Nodos

c) Recursos

d) Restricciones o relación entre los recursos

A continuación se detallan cada una de ellas.

a) Opciones de Configuración del Clúster

Se definen las opciones o propiedades que determinan el comportamiento del clúster como entidad grupal.

Algunas de las opciones son:

Tabla III. VII Opciones de configuración del Clúster

OPCIÓN	VALOR POR DEFECTO	DESCRPCIÓN
symmetric-cluster	true	Si es true, a los recursos se le permite correr en cualquier nodo. Si por el contrario es false, se tiene que especificar una restricción (ver Sección Restricciones) para indicar explícitamente dónde pueden ejecutarse.
stonith-enabled	true	Permite habilitar o deshabilitar Node Fencing.
stonith-action	reboot	Determina la acción del dispositivo stonith, los valores permitidos son: reboot o poweroff.
no-quorum-policy	stop	Acción a tomar cuando el clúster no tiene quórum. Los posibles valores son: • **ignore:** continúa la administración de todos los recursos • **freeze:** continúa la administración de los recursos, pero no recupera recursos desde nodos que están en la partición afectada. • **stop:** detiene todos los recursos en la partición del clúster afectada. • **suicide:** aislar todos los nodos de la partición del clúster afectada.
default-resource-stickiness	------	Establece si los recursos prefieren ejecutarse en el nodo actual o ser movidos a otro nodo en el cual se han ejecutado en mejores condiciones con anterioridad. Posibles valores son: • **0:** Es el valor neutro de tal manera que los recursos no tienen predilección por ejecutarse en un nodo o

		en otro. • **>0:** Los recursos prefieren permanecer en el nodo en el que actualmente se encuentran pero pueden ser movidos si otro nodo más conveniente se encuentra disponible. Cuanto mayor sea este valor mayor preferencia tendrá el recurso por mantenerse en el nodo en el que se encuentra actualmente. • **<0:** Los recursos prefieren moverse a otro nodo del clúster en lugar de permanecer en el que se encuentran. Cuanto menor sea este valor mayor será la preferencia de los recursos por abandonar el nodo en el que se encuentran actualmente ejecutándose. • **INFINITY:** Los recursos siempre permanecerán es el nodo actual a menos que este deje de estar operativo. • **–INFINITY:** Los recursos nunca se ejecutarán en el nodo, siempre intentarán moverse a otros nodos.
is-managed-default	true	Habilita la administración manual del clúster, lo que incluye operaciones como: mover los recursos de un nodo a otro, pararlos, arrancarlos, etc.
stop-orphan-resources	true	Elimina recursos que son detenidos. Esto puede ocurrir, por ejemplo, cuando el administrador elimina un recurso mediante una utilidad de administración del clúster sin que el recurso sea previamente parado. El recurso ya no es gestionado por el clúster, pero el proceso se encuentra ejecutándose en el nodo.
stop-orphan-actions	true	Elimina acciones que son canceladas.

b) Nodos

En esta sección se establecen los nodos que conforman el clúster. Cada nodo se identifica mediante el nombre de host y un identificador alfanumérico.

Para hacer más fácil la configuración de un clúster cuando el número de nodos comienza a no ser muy manejable, Heartbeat permite agrupar varios nodos en **grupos de nodos**. Así, de igual forma que es posible establecer restricciones o relaciones de orden, colocación o localización entre diferentes nodos lo es entre grupos de nodos.

Por ejemplo, podemos establecer para un determinado recurso que sólo sea posible ejecutarlo en un grupo de nodos determinado.

Los nodos y los atributos que describen los nodos se administran mediante el comando "node".

c) Recursos

Un recurso es la unidad básica a la que se puede dotar de alta disponibilidad. Un recurso es una abstracción, la cual puede ser de distintos tipos. Puede ser algo tan general como un servicio o tan concreto como un disco, un lector de tarjetas, o algo más abstracto como una dirección IP, reglas de un firewall, etc.

En muchas ocasiones, cuando el número de recursos en nuestro sistema de alta disponibilidad crece, es necesario tratar varios recursos de la misma forma, aplicar un gran número de restricciones o relaciones entre los recursos, etc. Es por ello que Heartbeat permite el establecimiento de **grupos de recursos**, siendo éstos manipulados como si de un solo recurso se tratara, facilitando enormemente las tareas de configuración y administración de la alta disponibilidad del clúster. Los recursos que forman parte de un grupo de recursos se caracterizan porque:

- Siempre son ejecutados en el mismo nodo, es decir, existe una restricción de colocación entre los recursos de un mismo grupo de forma que cada recurso siempre es ejecutado en el mismo nodo que el resto.
- Siempre son puestos en marcha y detenidos en el mismo orden, es decir, existe una restricción de orden entre los recursos de un mismo grupo que los obliga a ello.

Como los grupos de recursos son tratados como si fueran recursos individuales también es posible definir restricciones o relaciones entre grupos de recursos de la misma forma.

Comandos para recursos son:

- primitive
- monitor
- group
- clone
- ms/master (master-slave)

d) Restricciones o Relación entre los recursos

Como se ha mencionado anteriormente es posible establecer ciertas restricciones o relaciones entre nodos o grupos de nodos y entre recursos o grupos de recursos con el objetivo de definir de la forma más concisa posible su comportamiento.

Aplicar restricciones se hace necesario cuando existen dependencias entre recursos. Por ejemplo, el recurso Asterisk requiere ser ejecutado en el mismo nodo que el recurso IPaddr (IP virtual), ya que de lo contrario Asterisk estaría en funcionamiento pero no accesible para el establecimiento de llamadas.

Existen tres tipos de restricciones disponibles:

- ✓ **Restricciones de lugar.-** Definen en qué nodos puede ejecutarse un recurso.
- ✓ **Restricciones de colocación.-** Indican al clúster qué recursos corren en el mismo nodo y cuáles no pueden hacerlo.
- ✓ **Restricciones de orden.-** Estas restricciones son utilizadas para especificar el orden en que las operaciones start y stop que debe realizar un agente de recurso tienen que llevarse a cabo.

3.6.5. RSYNC

Rsync es una aplicación libre que permite sincronizar archivos y directorios entre máquinas de una red o entre dos ubicaciones en una misma máquina, minimizando el volumen de datos transferidos. Una característica importante de rsync no encontrada

en la mayoría de programas o protocolos es que la copia toma lugar con sólo una transmisión en cada dirección.

Algunas características de rsync son:

- ✓ Puede actualizar todo el árbol de directorios y el sistema de archivos.
- ✓ Opcionalmente conserva enlaces simbólicos, propietario, permisos, etc.
- ✓ Puede usar rsh, ssh o sockets para el transporte.
- ✓ Soporta rsync anónimo (autenticación usando el demonio rsync), lo cual es ideal para mirroring.

Con el uso de rsync solamente se copiarán los datos que se diferencian del origen con la copia de destino, con lo cual se evita tener que copiar de nuevo datos ya existentes.

Hay dos maneras diferentes para que rsync se ponga en contacto con un sistema remoto: el uso de un programa para conexión remota (por ejemplo, ssh o rsh) o ponerse en contacto con un demonio rsync.

Rsync se refiere a la parte local como el "cliente" y la parte remota como el "servidor". No hay que confundir "servidor" con un demonio rsync; un demonio siempre es un servidor, pero un servidor puede ser un demonio o un shell remoto.

CAPÍTULO IV

IMPLEMENTACIÓN DEL PROTOTIPO

4.1. ANÁLISIS DE REQUERIMIENTOS

Para la presente investigación se necesita implementar un prototipo de clúster que permita observar el comportamiento de un Call Center, para lo que se ha dispuesto la siguiente configuración en Hardware y Software.

4.1.1. Hardware

Se utilizó una configuración de tres nodos con las siguientes características:

Tabla IV. VIII Hardware utilizado

	Procesador	**Disco Duro**	**Memoria**	**Tarjeta de Red**
Nodo 1	Intel Core Duo 2,1 GHz	20 GB	1000 MB	2 NIC 10-100
Nodo 2 y 3	Intel Core Duo 2,1 GHz	20 GB	512 MB	2 NIC 10-100

4.1.2. Software

El hardware anteriormente descrito se virtualizó mediante VMware Workstation versión 7.1.

El sistema operativo utilizado en cada nodo junto con su configuración de software se detalla en la siguiente tabla.

Tabla IV. IX Software utilizado

PAQUETE	VERSIÓN
Sistema Operativo	
CentOS	5.5
VoIP	
Elastix	2.0.3
Modulo de Call Center	2.0.0
Clúster de Alta Disponibilidad	
Cluster-Glue	1.0.6
Resource- Agents	1.0.4
Heartbeat	3.0.3
Pacemaker	1.0.11
DMC (GUI)	0.9.7
Rsync	2.6.8

4.2. CONFIGURACIÓN DEL CALL CENTER

Previo las configuraciones que posteriormente se detallan, en cada uno de los nodos se instaló CentOS y Elastix.

El módulo de Call Center de Elastix está implementado alrededor del soporte de colas de Asterisk, el diseño del mismo asume que cada una de las colas alberga a uno o más agentes.

Con este módulo es posible la gestión de agentes, cola de llamadas, campañas entrantes y salientes, desarrollo de formularios, llamadas predictivas y tiempos de descanso. Cuenta además con una serie de estadísticas básicas y opciones de monitorización en tiempo real, tanto de los agentes como de las llamadas.

Se usa el término "campaña" para designar la ejecución de un conjunto de llamadas atendidas por un grupo de agentes, durante un intervalo de fechas específico.

La configuración a realizarse tiene el objetivo de generar llamadas de manera automática a números que previamente han sido subidos desde un archivo csv; lo cual se lleva a cabo mediante la creación de una campaña saliente.

Al activarse la campaña, y si hay agentes registrados como presentes en la cola asignada, el sistema marca los números telefónicos, tanto como agentes libres se encuentren en la cola.

A continuación se describen los pasos realizados para cumplir con lo antes mencionado.

4.2.1. Configurar el Módulo

Antes de empezar con las configuraciones, es necesario loguearse en el modulo de Call Center, para ello, en la interfaz web, ir a las pestaña “Call Center”, luego al menú “Configuración” y escribir los siguientes datos:

Login de asterisk :phpconfig
password de asterisk : php[onfig
password de asterisk(confirmation) : php[onfig

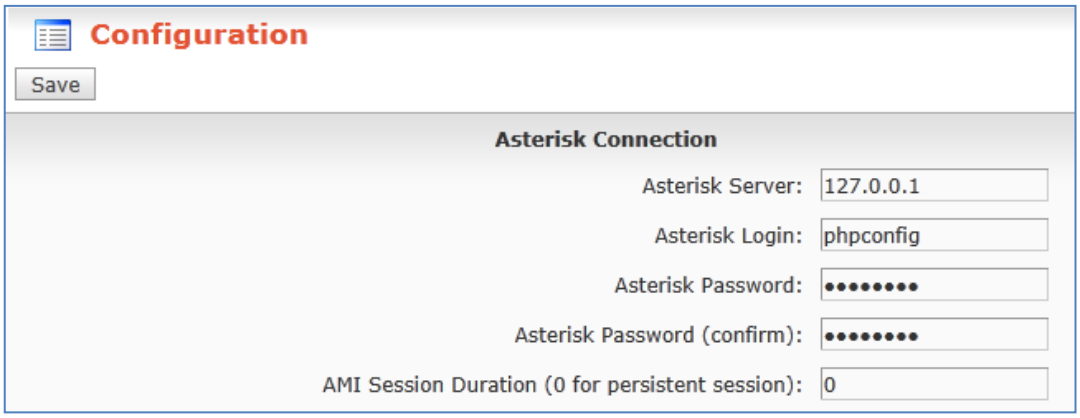

Figura IV. 25 Login en el Módulo de Call Center

Luego hacer click en "Guardar"; en el estado del dialer, hacer click en "Iniciar" para activar el servicio del marcador predictivo.

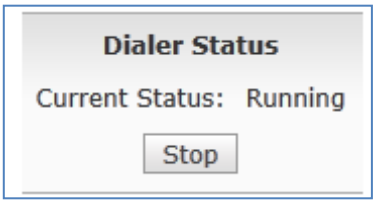

Figura IV. 26 Estado del Dialer

4.2.2. Crear Agentes

Con el menú Agentes es posible ingresar los datos de las personas que van a operar el sistema y que se les denomina agentes. Cada agente debe tener un número y clave asignado, para que por medio de esta se registre y reciba o realice llamadas.

Previo a asignar un número a los agentes, en la pestaña PBX se debe crear extensiones para ser utilizadas por los agentes.

Figura IV. 27 Datos para crear un agente

Una vez que se de click en “Guardar”, se desplegará una lista donde se puede observar el número, nombre y estado de cada uno de los agentes creados.

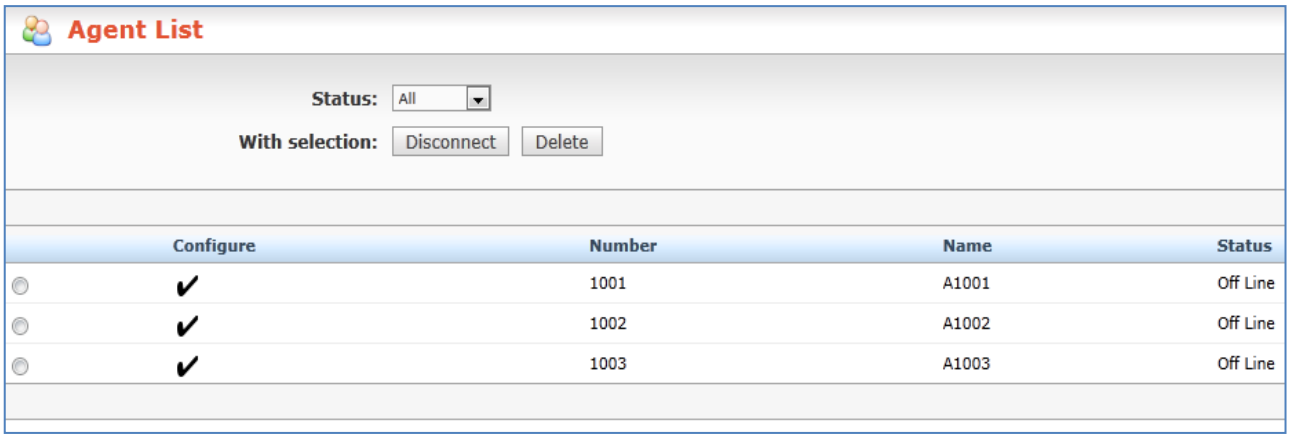

Figura IV. 28 Listado de Agentes

4.2.3. Crear Cola de Salida

Ir a la pestaña “PBX”, luego al menú izquierdo llamado “Colas” e ingresar los siguientes datos:

Queue Number: 10

Queue Name: COLAOUTPUT

Static Agents: A1001

Agent Timeout: 15 segundos

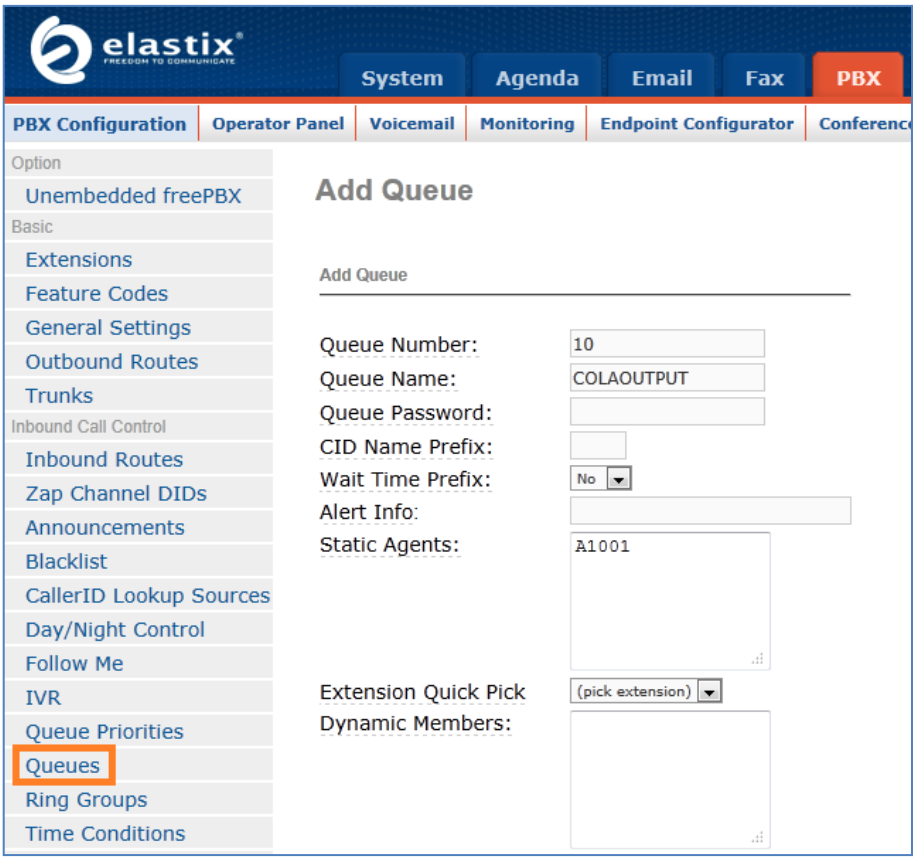

Figura IV. 29 Crear Cola

Mediante la variable Static Agent, se define cuales son los agentes que se registran en esta cola y los que realizaran las llamadas, para el presente caso es suficiente registrar un agente en la cola.

4.2.4. Crear Formulario

Los Formularios se crean con el objetivo de recolectar datos al momento de ejecutar una campaña.

Para crear un formulario, en la pestaña “Call Center”, ir al menú “Formularios” y luego click en “Crear Nuevo Formulario”. Una vez aquí, se debe ingresar el nombre del formulario y una breve descripción, además para cada uno de los campos se ingresa un nombre, orden y el tipo de datos que almacenará dicho campo, tal y como lo muestra la siguiente figura:

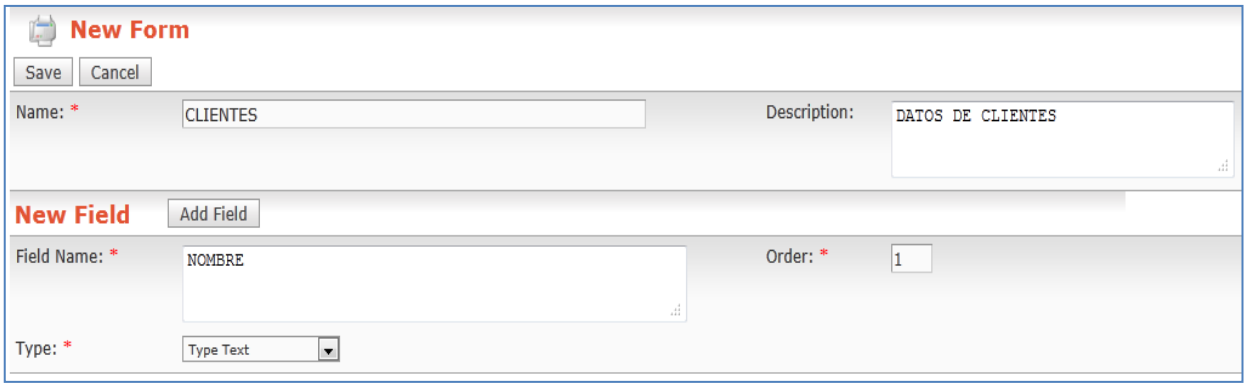

Figura IV. 30 Crear Formulario y campos

Es posible también observar una lista de formularios existentes, con el objetivo de hacer una vista preliminar de estos, antes de que sean usados en una campaña. Los datos que se pueden apreciar en el listado son los siguientes:

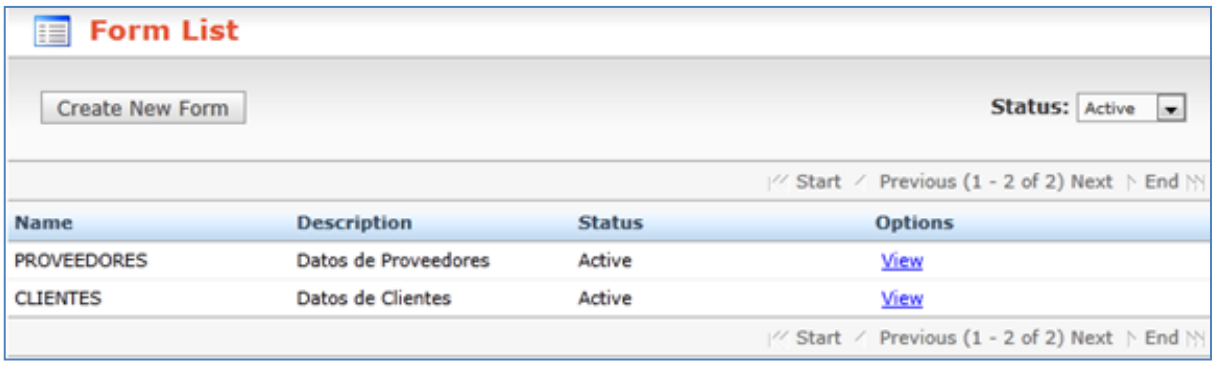

Figura IV. 31 Listado de Formularios

Cabe mencionar que para que un formulario pueda ser utilizado por una campaña su estado debe ser Activo.

4.2.5. Crear Campaña Saliente

La creación de la Campaña Saliente es el paso fundamental para que se generen las llamadas automáticamente, los datos a ingresarse son los siguientes:

Tabla IV. X Campos para crear una Campaña

CAMPO	DESCRIPCIÓN
Name	Nombre de la Campaña.
Range Date	Período de tiempo en que la campaña se va a encontrar vigente, luego de esta fecha no se realizan llamadas aun cuando hayan quedado números pendientes de marcar.
Schedule per Day	Horas del día en que entrará en funcionamiento la campaña.
Form	Seleccionar los formularios que se va a usar para recolectar datos.
Trunk	Troncal por donde salen las llamadas.
Context	Nombre del contexto que se utilizará, por defecto se usa from-internal.
Queue	Seleccionar la cola que se creó para ser utilizada por la campaña.
Retries	Número de veces que se intenta realizar una llamada que no se ha ejecutado con éxito.
Call File	Archivo con formato csv que contiene los números telefónicos a los cuales se realizará las llamadas.
Script	Es un mensaje de guía para los agentes en el momento que está en curso una llamada.

En el menú “Llamadas Salientes”, con la opción “Crear nueva Campaña” se despliega una pantalla como la que se muestra en la Figura 8, donde se ingresarán los datos explicados anteriormente.

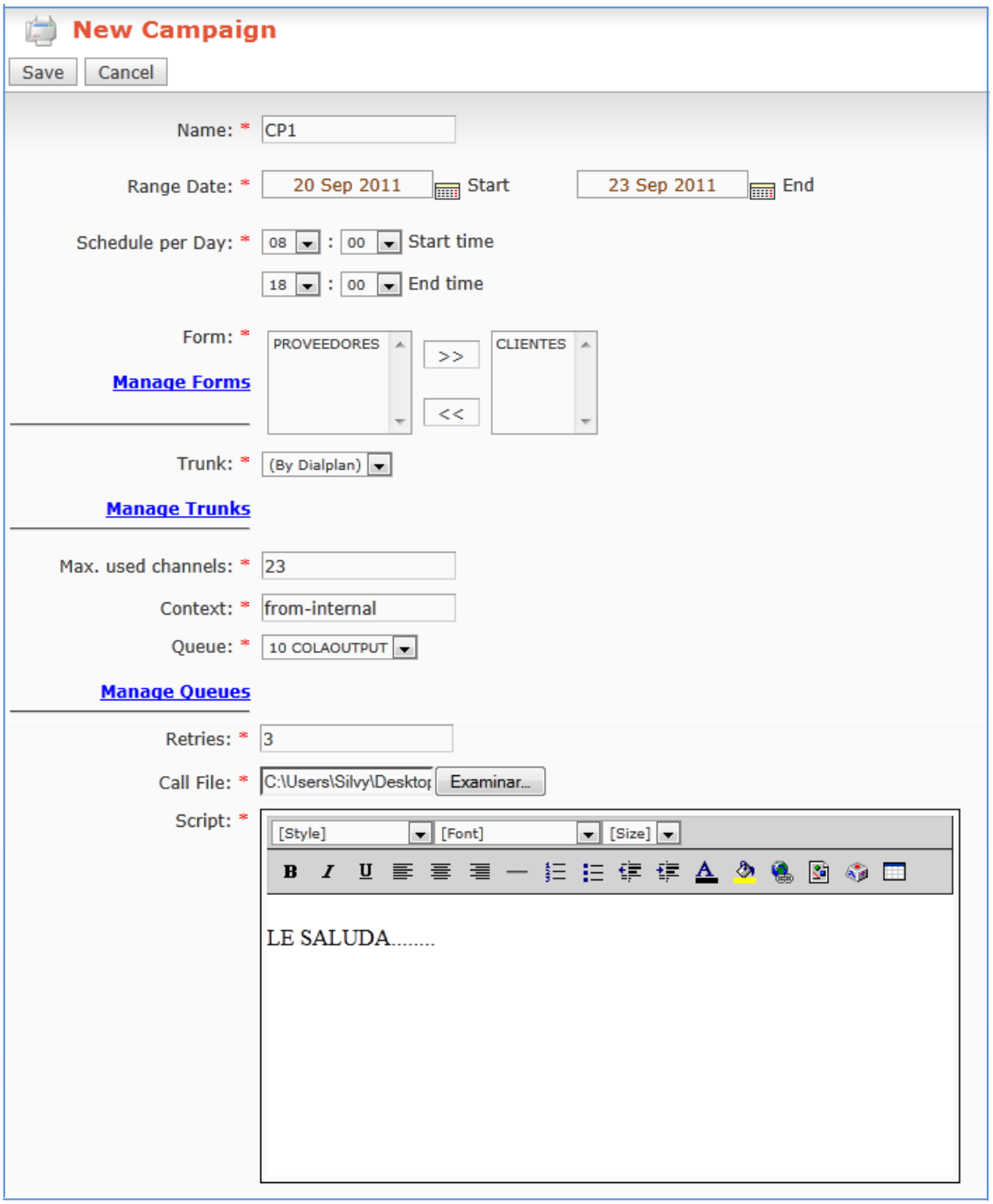

Figura IV. 32 Campaña Saliente

Por último al hacer click en guardar, aparecerá la campaña que acabamos de crear con estado de "Activa", es decir lista para ser ejecutada una vez que se encuentre logueado el agente denominado A1001.

4.2.6. Consola del Agente

Para ingresar al menú “Agent Console” es necesario proporcionar un número de agente con su respectiva extensión, ver Figura 9.

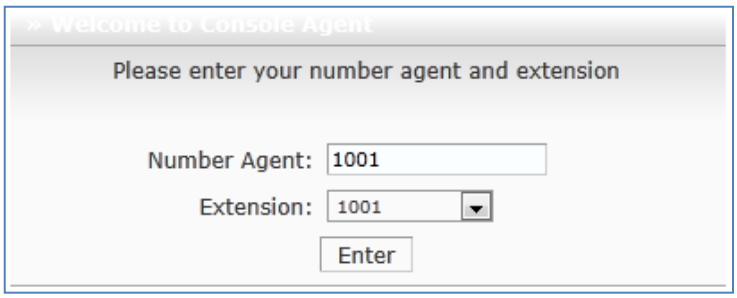

Figura IV. 33 Registro de Agente

Luego de pulsar en el botón ingresar, el anexo registrado como 1001 empezará a sonar y mediante un mensaje, solicitará que se digite la contraseña de agente seguido de la tecla #, lo que permitirá observar la Consola del Agente.

Transcurridos algunos segundos, en cuanto el marcador progresivo detecte que una llamada ha sido establecida, la consola se mostrará como en la Figura10.

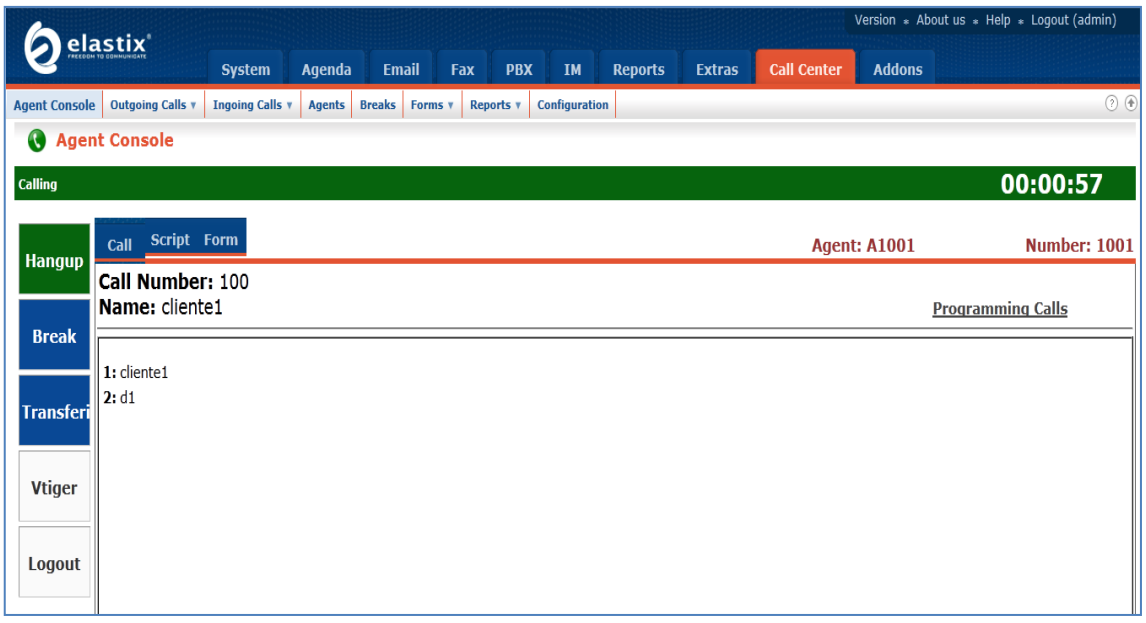

Figura IV. 34 Consola de Agente con llamada establecida

4.3. CONFIGURACIÓN DEL CLÚSTER DE ALTA DISPONIBILIDAD

En este apartado se detallan las configuraciones realizadas para el clúster de alta disponibilidad, así como también se explicará si las configuraciones se las debe realizar en todos los nodos del clúster o no.

En la siguiente figura se puede observar que cada servidor cuenta con dos interfaces de red, una para la comunicación entre los nodos del clúster (Ethernet 1) y la otra para brindar el servicio (Ethernet 0).

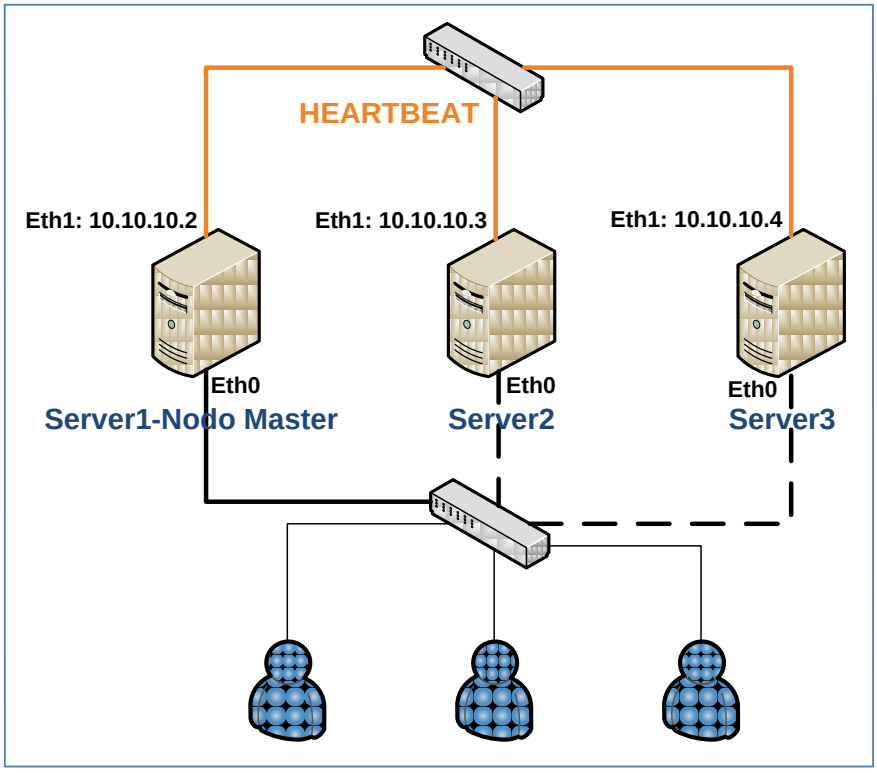

Figura IV. 35 Diagrama del Clúster de Alta Disponibilidad

4.3.1. Configuración de Heartbeat

Una vez instalado Heartbeat, el siguiente paso es editar 3 ficheros en los tres nodos, estos son:

- */etc/hosts*
- */etc/ha.d/ha.cf*

- */etc/ha.d/authkeys*

Y en el caso que se esté utilizando una versión de heartbeat anterior a 2.1.4, también se deberá editar */etc/ha.d/haresources.*

4.3.1.1. Nombre de los Nodos - fichero */etc/hosts*

Cada uno de los nodos debe disponer de un nombre único que lo identifica en el clúster, los mismos que serán registrados en el fichero */etc/hosts,* además de la dirección IP asignada a cada uno de ellos.

La salida del comando *cat* en el servidor uno (nodo master), es la siguiente:

```
[root@server1 ~]# cat /etc/hosts
# Do not remove the following line, or various programs
# that require network functionality will fail.
127.0.0.1               localhost.localdomain localhost
::1             localhost6.localdomain6 localhost6

10.10.10.2 server1
10.10.10.3 server2
10.10.10.4 server3
```

4.3.1.2. Fichero */etc/ha.d/ha.cf*

La configuración principal del clúster es la que se encuentra en este fichero, es aquí donde se define cada cuantos segundos se enviaran los latidos (ping) entre los nodos, después de que tiempo se considera un nodo muerto, el puerto que se utiliza para la comunicación, etc.

```
#
debugfile /var/log/ha-debug
#
logfile    /var/log/ha-log
#
logfacility     local0
#
keepalive 2
#
deadtime 30
#
warntime 10
#
initdead 120
```

```
#
udpport    694
#
bcast eth1
#
ucast eth1 10.10.10.3
#
node server1
node server2
node server3
#
crm respawn
```

A continuación se explica los principales parámetros de este fichero:

Tabla IV. XI Parámetros del fichero ha.cf

PARÁMETRO	FUNCIÓN
debugfile	Permite especificar el fichero donde se guardarán los mensajes de depuración.
logfile	Fichero donde se guarda el resto de mensajes.
logfacility	Indica la facilidad para el uso de syslog() o logger.
keepalive	Intervalo de tiempo (en segundos) entre ping y ping hacia cada nodo del clúster.
deadtime	Tiempo que debe transcurrir para asumir que un nodo no se encuentra disponible o que está muerto.
warntime	Tiempo que transcurre antes de indicar la advertencia de último intento de ping en la monitorización.
initdead	Tiempo obligatorio que el nodo tiene que esperar antes de iniciar Heartbeat. Si uno de los nodos del clúster no arranca antes de que expire este tiempo se considerará un nodo “muerto”.
udpport	Puerto que utiliza Heartbeat para las comunicaciones unicast y broadcast, tanto para enviar como recibir los ping.
bcast	Interfaz de red que usa Heartbeat para el envío de pings.
ucast	Interfaz utilizada para la comunicación entre pares de nodos en el clúster. Además de la interfaz se especifica la dirección IP del otro nodo, por lo que este parámetro es diferente en cada nodo.

node	Especifica los nodos que forman parte del clúster.
crm	Permite el uso o no del administrador de recursos del clúster.

4.3.1.3. Fichero */etc/ha.d/authkeys*

Este fichero contiene información que Heartbeat utiliza para autenticar a los miembros del clúster. Por seguridad deberá tener exclusivamente permisos de lectura/escritura para el usuario root.

Se puede seleccionar entre tres métodos de autentificación:

- **crc,** este método es recomendable cuando los nodos del clúster se comunican a través de una red segura. Crc apenas consume recursos por lo que su aplicación también es recomendable cuando necesitamos disponer de la máxima cantidad de recursos posible.
- **sha1** es justamente lo contrario, es el método que requiere una mayor cantidad de recursos de CPU en su aplicación al mismo tiempo que aporta una gran seguridad.
- **md5,** si por el contrario, se desea una buena relación seguridad/consumo CPU, se puede utilizar md5.

El formato del fichero es el siguiente:

```
auth <identificador>
<identificador> <metodo_autenticacion> [<clave_autenticacion>]
```

Una vez finalizada la configuración de estos ficheros y previo a la configuración de pacemaker se debe iniciar el servicio Heartbeat con el siguiente comando:

```
service heartbeat start
```

4.3.2. Configuración de Pacemaker

Continuando con la configuración del clúster, es momento de definir los recursos que se van a dotar de alta disponibilidad (en este caso el servicio asterisk, aquel que permite el funcionamiento del Call Center), así como en que nodos deben ser ejecutados, en qué orden, etc.

Para configurar Pacemaker se usa la herramienta crm. Las instrucciones pueden introducirse en cualquier nodo, y serán propagadas automáticamente por todo el clúster.

4.3.2.1. Recursos

Los recursos que se han configurado en el Clúster son los siguientes:

- ✓ IP Virtual
- ✓ Asterisk
- ✓ Mysqld
- ✓ Elastixdialer

IP Virtual

El primer recurso a ser creado es la dirección virtual, que será la encargada de mantener el servicio de telefonía siempre disponible para los clientes.

Por ejemplo, si un nodo que está ejecutando Asterisk falla y no se dispone de una IP virtual migrar el servicio a otro nodo no sirve de nada ya que los clientes siguen intentando acceder al servicio a través de la dirección IP del nodo que ha fallado; es entonces cuando aparece la necesidad de utilizar una dirección IP virtual, tratada como un recurso más en el clúster y que es migrado igualmente entre los nodos, permitiendo que los servicios se encuentren siempre accesibles.

La sintaxis para la creación de un recurso es la siguiente:

```
primitive <rsc> [<class>:[<provider>:]]<type>
        [params <param>=<value> [<param>=<value>...]]
        [meta <attribute>=<value> [<attribute>=<value>...]]
        [operations id_spec
            [op op_type [<attribute>=<value>...] ...]]
```

Siendo así la línea de comandos para crear la ip virtual es:

```
crm configure primitive vip ocf:heartbeat:IPaddr params ip=172.30.104.240 nic=eth1 cidr_netmask=24 op monitor interval=10s
```

Donde:

- crm configure primitive.- Permite añadir el recurso
- vip ocf:heartbeat:IPaddr.- El nombre del recurso es "vip" y hace uso del RA (resource agent, o script que controla el recurso) "ocf:heartbeat:IPaddr".
- Los parámetros son la IP, la interfaz que atenderá a esa IP y la máscara de red.
- Como opciones, el "monitor interval" concreta la frecuencia con la que Pacemaker se asegura de que el recurso está funcionando.

Asterisk

Este recurso consiste en la ejecución del servicio Asterisk para el funcionamiento del módulo de Call Center proporcionado por Elastix, en el clúster.

El recurso se creó mediante:

```
crm configure primitive voip lsb:asterisk op monitor interval=15s
```

Como se puede notar en la línea de configuración, el agente de recurso para asterisk es de tipo LSB, por lo tanto para que sea Heartbeat y no el propio Sistema Operativo el que maneje las operaciones start y stop, es necesario realizar lo siguiente:

- Desactivar el inicio automático de asterisk cuando arranca el Sistema Operativo.
- Crear un Enlace Simbólico en el directorio donde constan los recursos controlados por Heartbeat.

  Una vez ubicados en **/etc/ha.d/resource.d/,** dicho enlace se creó con **/etc/init.d/asterisk asterisk.**

Grupo de Recursos

Debido a que el recurso Asterisk precisa que la dirección IP virtual esté disponible para su funcionamiento, estos dos recursos deben conformar un grupo de recursos.

Luego que dichos recursos han sido agrupados se cumple que: los recursos se ejecutan siempre en el mismo nodo y también que los recursos deben ser iniciados y detenidos siempre en el mismo orden (primero la IP virtual, después Asterisk), es decir restricción de colocación y de orden respectivamente.

El comando utilizado para crear el grupo de recursos es:

```
crm configure group recursos vip voip
```

Donde “recursos” es el nombre elegido para el grupo recursos, “vip” y “voip” son los identificadores de los recursos antes creados y los cuales se desea conformen el grupo.

Además, para que la sincronización de los datos pueda reflejarse en la interfaz web de elastix, fue necesario incorporar dos servicios más al grupo de recursos, estos son mysqld y elastixdialer, los mismos que al igual que la dirección virtual y asterisk pueden ser creados a través de comandos, pero por demostración se utiliza la interfaz grafica.

La pantalla que indica los servidores activos y el grupo de recursos configurado hasta el momento, se muestra de la siguiente manera:

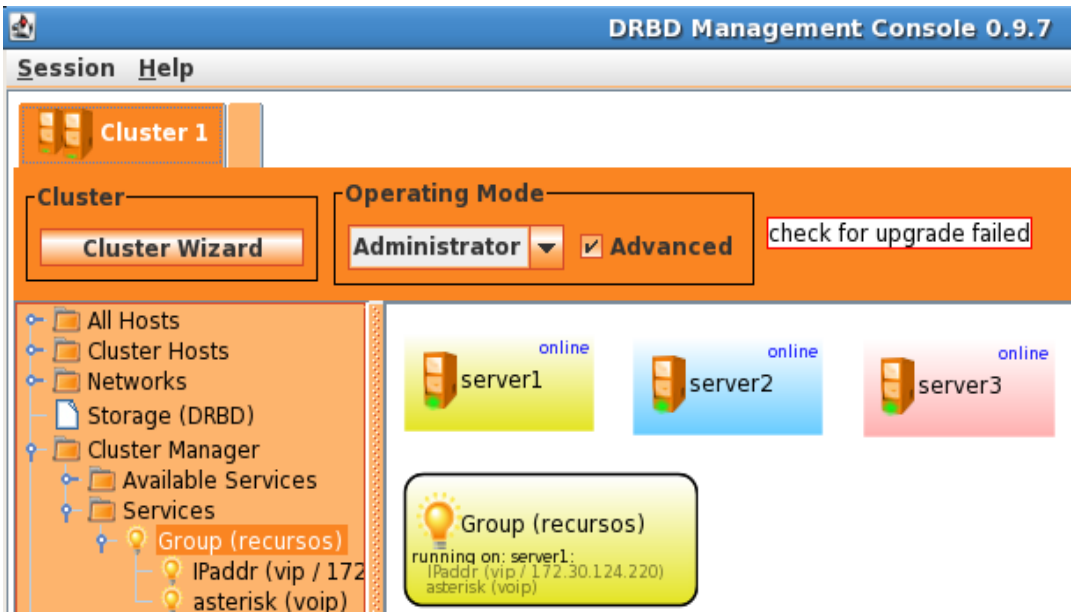

Figura IV. 36 Interfaz gráfica con los servidores y el grupo de recursos

Para crear un nuevo recurso, basta ubicarse sobre el gráfico del grupo de recursos, hacer click y seleccionar la opción “Add Group Service”, lo cual hará que se despliegue un submenú con el tipo de agente de recurso, donde se hará click en LSB y finalmente en mysqld (Ver figura 13).

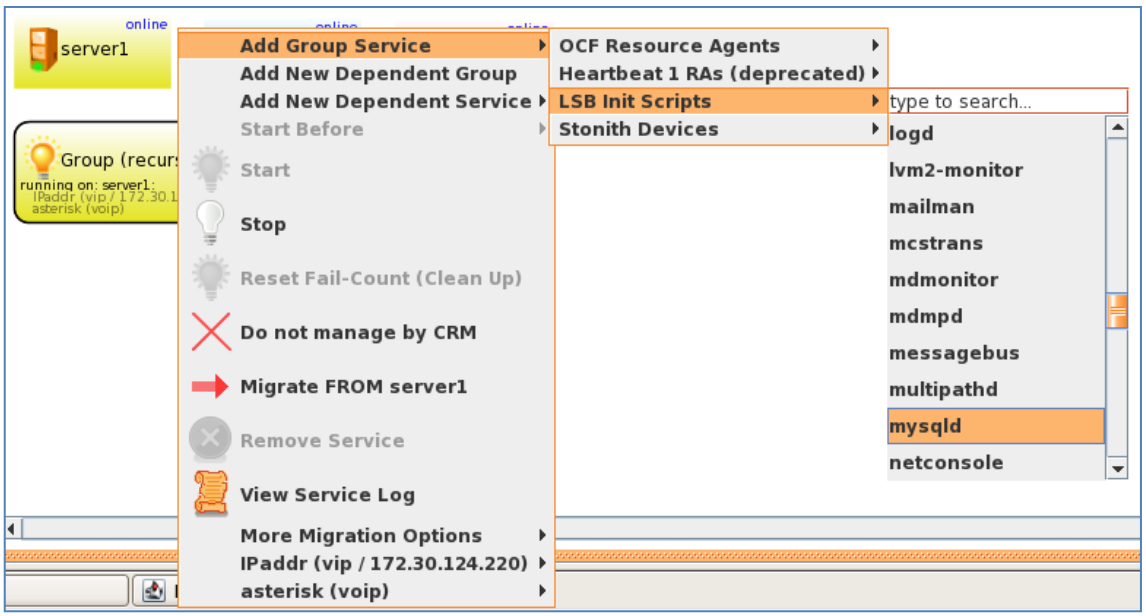

Figura IV. 37 Agregar recurso mysqld al grupo de recursos

En cuanto al servicio elastixdialer que es el que controla el marcador predictivo del modulo Call Center, se seguirán los mismos pasos ya que los dos servicios son de tipo LSB.

Para que los nuevos recursos arranquen, es necesario definir el tiempo de monitoreo y el tiempo en que dicha operación debe ser llevada a cabo (timeout), valores que también pueden ser establecidos desde la interfaz grafica a través del panel derecho, donde se puede encontrar las variables disponibles para cada recurso.

Los valores especificados para el intervalo de monitorización y para el timeout son 10 y 20 segundos respectivamente.

4.3.2.2. Opciones del Clúster

En este apartado, se explican algunas variables globales para el clúster.

Para modificar los valores de las propiedades del clúster se usa la palabra clave **property**, también en el nivel configure:

```
crm configure property expected-quorum-votes="2"
crm configure property no-quorum-policy=ignore
```

Con la primera línea se define el número de votos con los que el clúster tiene quórum y con la segunda se indica que en caso de no tener quórum, el clúster siga funcionando.

Para continuar, se establecen los valores para las propiedades ***default-resource-stickiness*** y ***default-resourcefailure-stickiness***, con las que se puede establecer la adherencia por defecto de los recursos para ser ejecutados en el nodo actual o por el contrario ser migrados a otro nodo que ofrece unas condiciones mejores. Se asigna una puntuación positiva en *default-resource-stickiness* para premiar la correcta ejecución del grupo en un nodo y una puntuación negativa en *defaultresource-failure-stickiness* para penalizar la ejecución del grupo en un nodo si han sido detectado fallos.

```
crm configure property default-resource-stickiness=20
crm configure property default-resource-failure-stickiness=-20
```

Como último paso se debe establecer los mecanismos por los cuales Heartbeat va a determinar si es necesario migrar los servicios del nodo activo a otro o no. Para ello se concreta la definición de las operaciones de monitorización de cada uno de los recursos individuales que forman el grupo *recursos* (*vip* y *voip*) y hacer uso de la propiedad *migration-threshold*.

Tanto para *vip* como para voip se establece el valor de la propiedad ***timeout*** para la operación ***monitor***, ya que permite fijar el tiempo máximo permisible para la ejecución de la operación. Así Heartbeat, una vez que transcurre este tiempo sin haber

culminado la operación correctamente, determina que algo debe ir mal en la ejecución/disponibilidad del recurso. Los timeouts para las operaciones deben ser al menos de la misma duración que las recomendaciones disponibles en *meta-data* (20 segundos), aunque eso no quiere decir que para un recurso determinado sea suficiente.

Con los timeouts establecidos para la monitorización de los dos recursos se procede a cambiar la propiedad ***migration-threshold*** para el grupo de recursos.

Esta propiedad permite fijar el número de fallos o errores que pueden tener lugar para el grupo de recursos. Una vez alcanzado este límite, el grupo de recursos es migrado a otro nodo.

```
crm resource meta recursos set migration-threshold 1
```

4.3.3. Configuración de Rsync

Una vez que se ha configurado la alta disponibilidad del servicio es necesario también usar rsync para la sincronización de los directorios con los que trabaja asterisk, logrando de esta manera que la información sea igual entre los nodos del clúster.

Como primer paso se creó un script que ejecute las instrucciones de sincronización, el cual contiene lo siguiente:

```
#!/bin/bash

# Este script hace uso de rsync para sincronizar
# ficheros entre los nodos del cluster.

# El script esta pensado para ejecutarse cada 30 seg. (crontab)

# Ademas, es muy importante que las horas esten sincronizadas.
#   En caso contrario, el comportamiento es ciertamente
problematico.

# Los nombres de cada maquina deben ajustarse a `uname -n`
```

```
# No importa de donde venga la lista de nombres mientras se
cumpla.
# Debe haber un espacio en blanco entre cada nombre de maquina.

#  Se comprueba un lock file para que no se superpongan
ejecuciones de este script

#########################
# Variables             #
#########################
LISTA_NODOS="server1 server2 server3"
THIS=`uname -n`
THIS_PID=$$
LOCK_FILE="/var/run/sync_files.sh.LOCK"

#########################
# Programa              #
#########################

# Validando ejecucion
if [ -e $LOCK_FILE ]
then
     if [ "`cat $LOCK_FILE`" != $THIS_PID ]
        then
             exit 1
        fi
else
     echo "$THIS_PID" > $LOCK_FILE
fi

# Sincronia
for nodo in $LISTA_NODOS
do
   # Si el nodo de la lista no es el propio nodo ejecutando este
script
        # se ejecuta la instruccion de rsync
        if [ "$nodo" != "$THIS" ]
        then
                # Se sincronizan los ficheros especificados
                # Usando como copia maestra el mas reciente.
                echo $nodo
           rsync -uavz $nodo::asterisk /etc/asterisk/
           rsync -uavz $nodo::asterisklib /var/lib/asterisk/
           rsync -uavz $nodo::asteriskspool /var/spool/asterisk/
           rsync -uavz $nodo::asteriskmysql /var/lib/mysql/
           rsync -uavz $nodo::elastixhtml /var/www/html/
           rsync -uavz $nodo::elastixdb /var/www/db/
           rsync -uavz $nodo::elastixdialer /opt/elastix/dialer
        fi
done

# Finalizando ejecucion
rm -f $LOCK_FILE
```

Las opciones de rsync usadas indican:

-a, modo archivo, igual que si se usaran los parámetros **-rlptgoD**

-r, para que el recorra toda la estructura de directorios que se indique

-l, para que copie enlaces simbólicos como enlaces simbólicos

-p, para que mantenga los permisos

-t, para que se mantenga la hora del fichero

-g, para que se mantenga el grupo

-o, para que se mantenga el propietario, y

-D, para que se mantengan los ficheros de dispositivo (sólo para root).

-u, update, mantiene el archivo en el destino, si es posterior.

-v, muestra información durante la transferencia.

-z, comprime los datos durante la transferencia.

Además se necesita del fichero rsyncd.conf, ubicado bajo /etc, cuyo contenido es el que sigue:

```
motd file = /etc/rsyncd.motd
log file = /var/log/rsyncd.log
pid file = /var/run/rsyncd.pid
lock file = /var/run/rsync.lock

uid = nobody
gid = nobody

[asterisk]
   path = /etc/asterisk/
   comment = asterisk
   read only = no
   list = yes
   #auth users = root
   secrets file = /etc/rsync/pass.conf

[asterisklib]
   path = /var/lib/asterisk/
   comment = asterisklib
   read only = no
   list = yes
   #auth users = root
   secrets file = /etc/rsync/pass.conf

[asteriskspool]
```

```
   path = /var/spool/asterisk/
   comment = asteriskspool
   read only = no
   list = yes
   #auth users = root
   secrets file = /etc/rsync/pass.conf

[asteriskmysql]
   path = /var/lib/mysql/
   comment = asteriskmysql
   read only = no
   list = yes
   #auth users = root
   secrets file = /etc/rsync/pass.conf

[elastixhtml]
   path = /var/www/html/
   comment = elastixhtml
   read only = no
   list = yes
   #auth users = root
   secrets file = /etc/rsync/pass.conf

[elastixdb]
   path = /var/www/db/
   comment = elastixdb
   read only = no
   list = yes
   #auth users = root
   secrets file = /etc/rsync/pass.conf

[elastixdialer]
   path = /opt/elastix/dialer/
   comment = elastixdialer
   read only = no
   list = yes
   #auth users = root
   secrets file = /etc/rsync/pass.conf
root:root
```

Para ejecutar rsync en modo demonio el comando es:

```
rsync --daemon
```

Con el objetivo que el script sea ejecutado automáticamente, se agregaron las siguientes líneas a */etc/crontab:*

```
* * * * *   root   /etc/rsync/sync_files.sh
* * * * *   root   sleep 30 && /etc/rsync/sync_files.sh
```

CAPÍTULO V

MONITOREO DEL CLUSTER

5.1. FUNCIONAMIENTO

El Clúster de Alta Disponibilidad está formado por tres nodos, cada uno de ellos con Asterisk instalado. Se estableció una configuración en la que un nodo, llamado activo (server1), posee y ejecuta el recurso Asterisk mientras que los otros nodos (server2 y server3) se mantienen a la espera de posibles fallos (configuración Activo/Pasivo).

Mediante Heartbeat los nodos se comunicarán constantemente, de forma que uno puede monitorizar el estado del otro (y viceversa). Además, cada nodo monitoriza en qué estado se encuentra el recurso, para la detección de posibles irregularidades y fallos.

Cuando un fallo sea detectado en server1, ya sea un fallo hardware o un error en la ejecución de asterisk, este recurso es migrado del primer nodo al segundo, que pasa a ser consecuentemente el nuevo nodo activo (server2), el mismo que para continuar ofreciendo el servicio Asterisk toma la dirección IP Virtual asociada al servicio.

Ante situaciones de fallo similares el sistema de alta disponibilidad actúa de forma análoga, consiguiendo que cuando se produzcan fallos en la ejecución del servicio y aunque éste deje de estar disponible durante unos instantes, el tiempo que transcurre hasta ejecutar el servicio de nuevo en otro nodo sea mínimo, normalmente inapreciable por los usuarios.

Es así como es posible recuperarse ante un fallo en la ejecución del servicio de forma limpia y rápida; cuando el administrador detecte el error y lo solucione, el nodo (anteriormente activo) pasa a encontrarse de nuevo disponible para prestar sus servicios al clúster. Ahora, dependiendo de la configuración realizada en Heartbeat, el servicio asterisk que ahora se encuentran en otro nodo pueden volver al nodo original o bien permanecer en el que actualmente es activo.

Independientemente de la recuperación ante desastres que permite Heartbeat, los recursos también pueden ser migrados por Heartbeat de forma manual, ya que ocasionalmente es necesario realizar actividades de administración y mantenimiento (actualizaciones, instalaciones, reparación y sustitución de componentes, etc.) de los nodos que ejecutan los servicios críticos, aunque no haya sido detectado ningún error.

5.2. PRUEBAS DE LA INFRAESTRUCTURA DE HA

Para probar la configuración establecida para el clúster de alta disponibilidad se ha considerado analizar el comportamiento del mismo ante dos sucesos diferentes que provocan la migración del grupo de recursos de un nodo a otro, estos son: apagado del nodo activo y caída del servicio Asterisk.

5.2.1. Apagado del Nodo Activo

Con el apagado del nodo activo en el que se encuentra actualmente el grupo de recursos en ejecución se puede simular algunas situaciones reales como la pérdida de

suministro eléctrico en el servidor o un apagado del sistema forzado por motivos fundados en operaciones de mantenimiento.

En la siguiente figura se puede observar cada uno de los nodos y su estado (en este caso online los tres nodos), también se observa que los servicios del grupo de recursos esta ejecutándose.

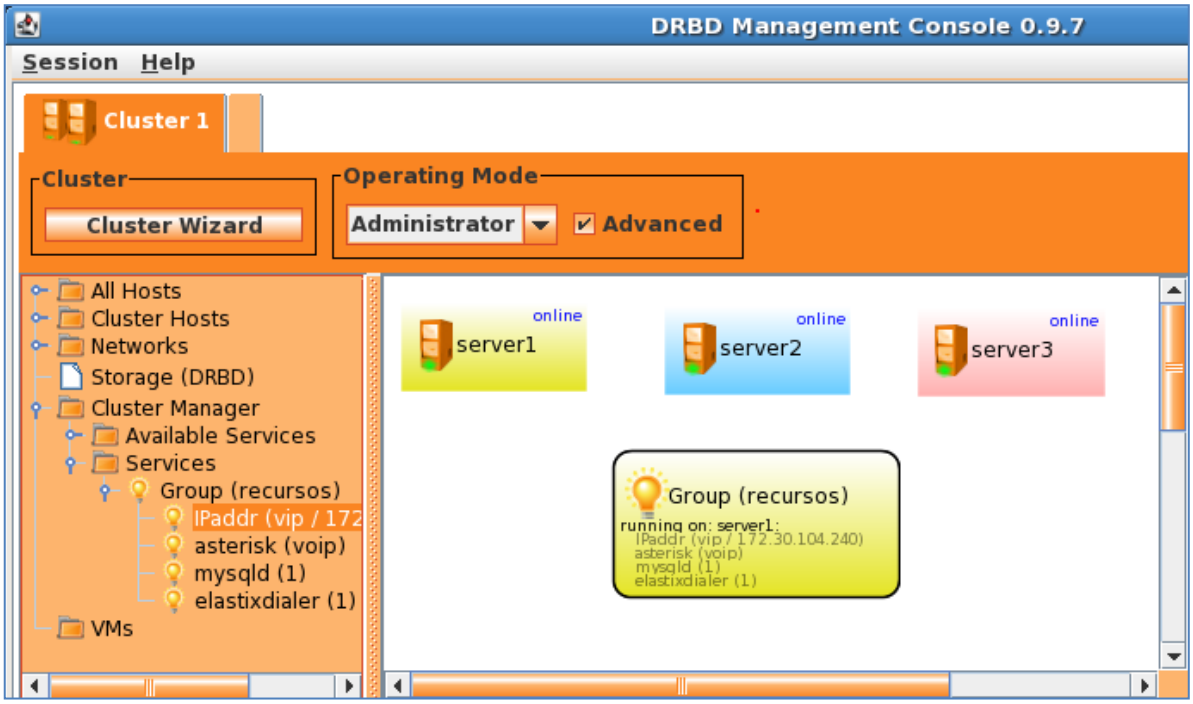

Figura V. 38 Grupo de Recursos en ejecución

Para comprobar que efectivamente la dirección virtual se ha iniciado en el Nodo 1, se ejecuta el comando ifconfig (Ver Figura 39) y se observa que en realidad la dirección esta activa.

En cuanto a asterisk, es suficiente con conectarse a la consola del servicio ejecutando **asterisk –r**, apareciendo el command line interfaces de Asterisk (CLI) (Ver Figura 40).

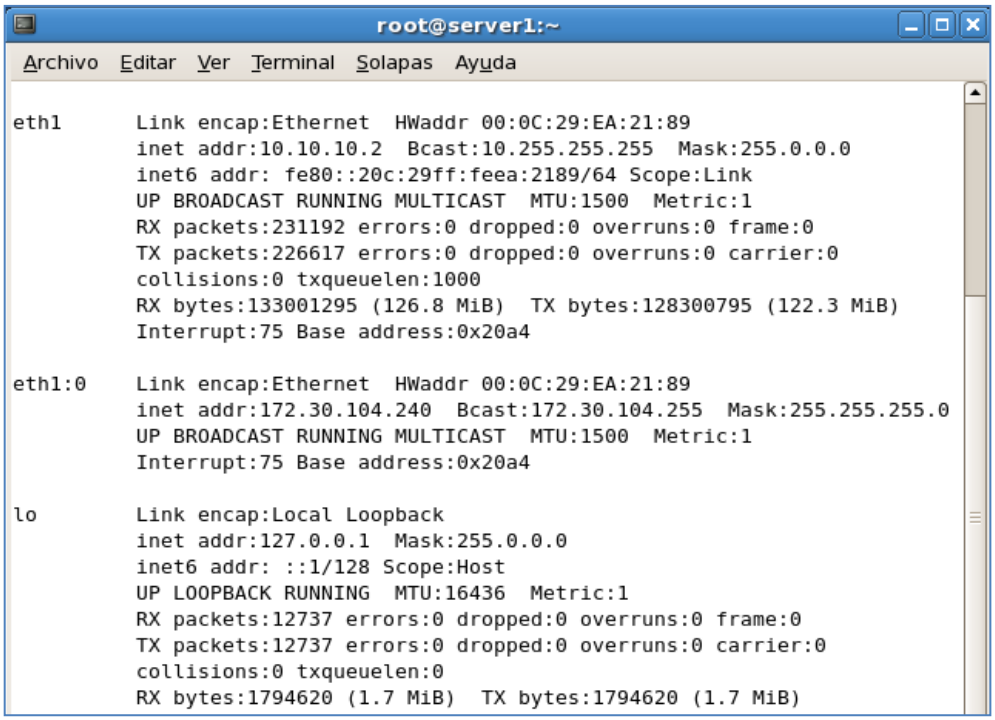

Figura V. 39 ifconfig en Nodo 1

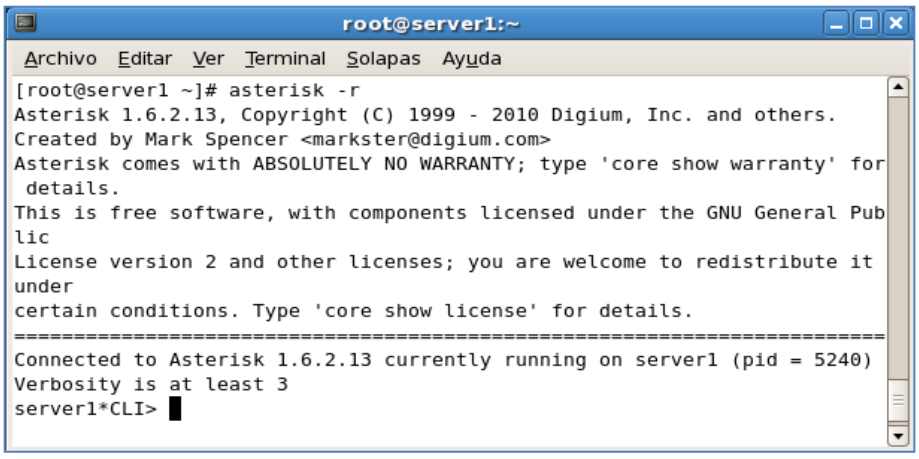

Figura V. 40 CLI de asterisk en el Nodo 1

En el Nodo 2 y 3, en cambio, se dispone solamente de la dirección de red 10.10.10.3 y 10.10.10.4 respectivamente (no la dirección IP virtual) y asterisk no se encuentra accesible (el servicio no es iniciado en estos nodos).

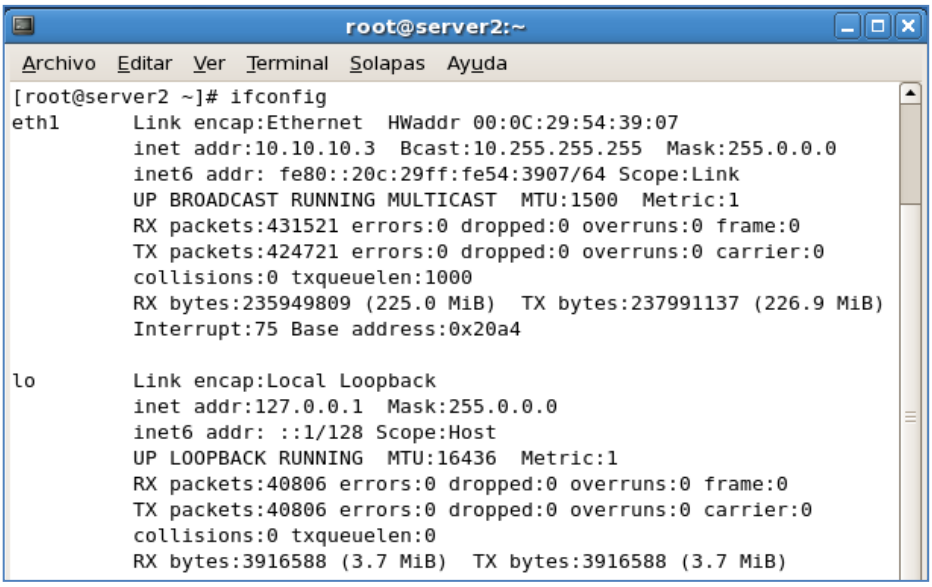

Figura V. 41 ifconfig en Nodo 2

Figura V. 42 CLI de asterisk en el Nodo 2

Ahora se procede con el apagado del nodo y a través del log de Heartbeat en el Nodo 2 se puede apreciar cómo este detecta la caída del Nodo Activo.

```
server2 heartbeat: [4278]: WARN: node server1: is dead
server2 heartbeat: [4278]: info: Link server1:eth1 dead.
server2 crmd: [4715]: notice: crmd_ha_status_callback: Status
update: Node server1 now has status [dead] (DC=false)
server2 crmd: [4715]: info: crm_update_peer_proc: server1.ais is
now offline
```

Posteriormente el grupo de recursos es arrancado en el siguiente nodo disponible, en este caso en el Nodo 2 (Ver Figura 43), luego haciendo uso del comando **crm status** se comprueba el estado del Nodo 1 (offline) y que cada uno de los servicios esta iniciado y funcionando en el Nodo 2 (Ver Figura 44).

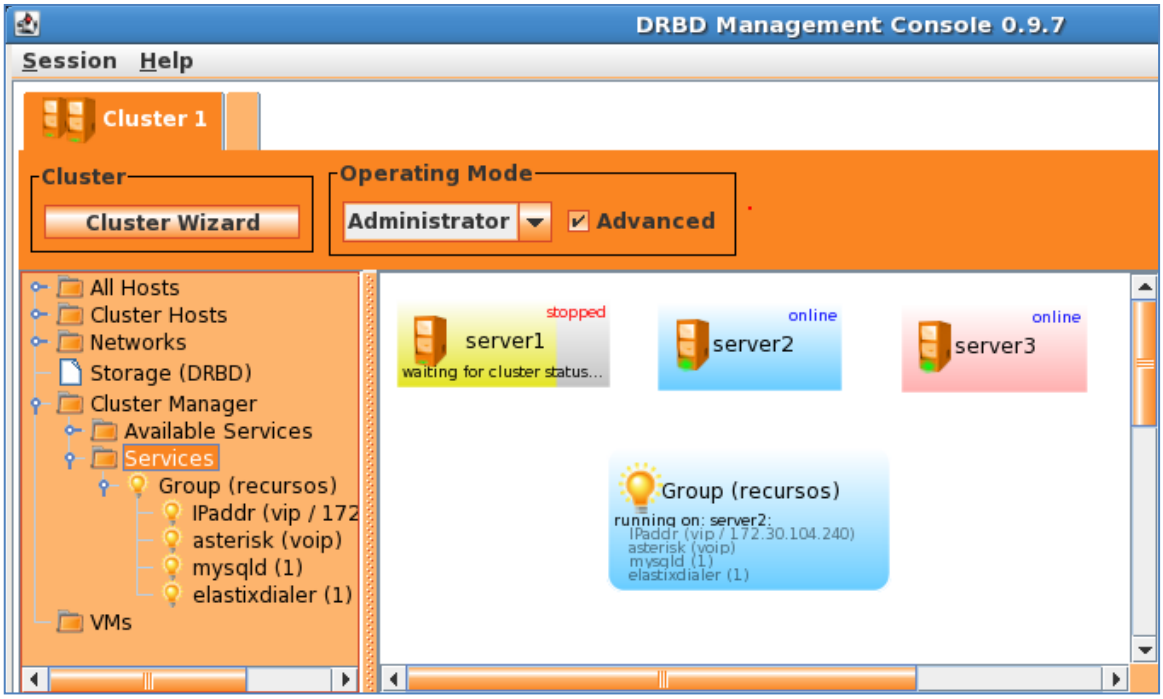

Figura V. 43 Ejecución de Recursos en el Nodo 2

Figura V. 44 Resultados del comando **crm status**

5.2.2. Caída del Servicio

En esta prueba se provoca la caída del servicio asterisk y se observa como Heartbeat reacciona ante este evento, el cual depende del umbral de migración establecido para este recurso, como se explicó en el capítulo anterior, el valor de *migration-threshold* se fijó en uno lo cual quiere decir que ante el primer fallo de asterisk se debe migrar el recurso.

A continuación, para simular la caída del servicio se procede a detenerlo manualmente, y Heartbeat lo detecta al consumirse el timeout (20 segundos) asociado a la operación monitor del recurso *voip,* y como el límite de fallos ha sido alcanzado, el grupo de recursos pasará a estar activo en el Nodo 3. (Ver Figura 45).

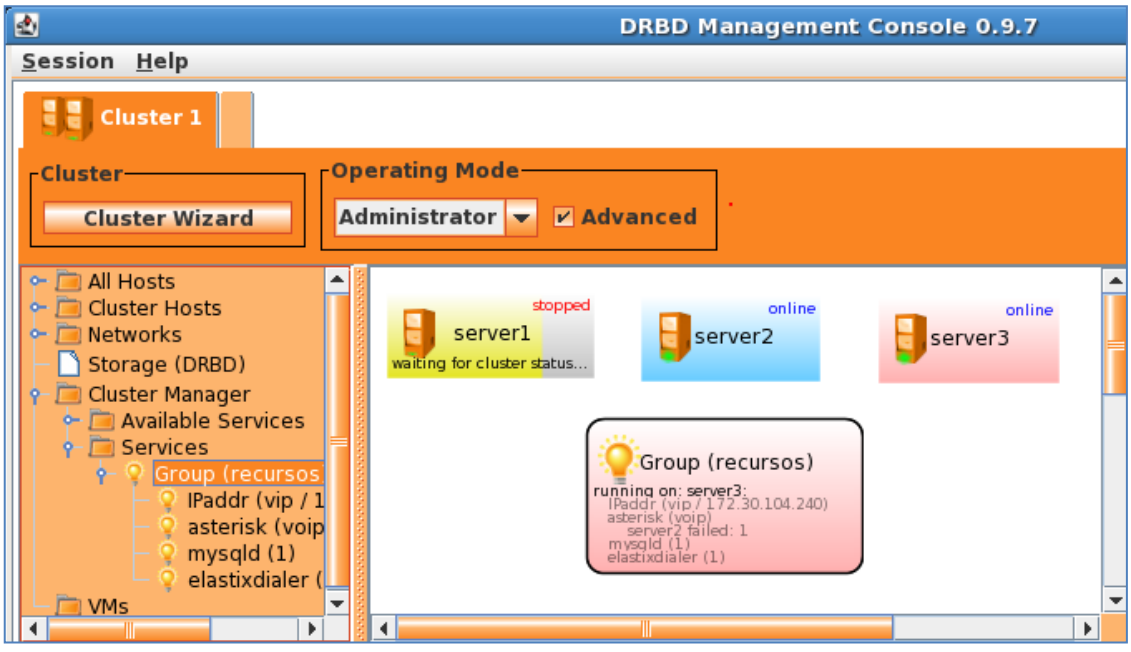

Figura V. 45 Ejecución de Recursos en el Nodo 3

Además se puede apreciar como el fallo del recurso asterisk en el Nodo 2 ya es registrado

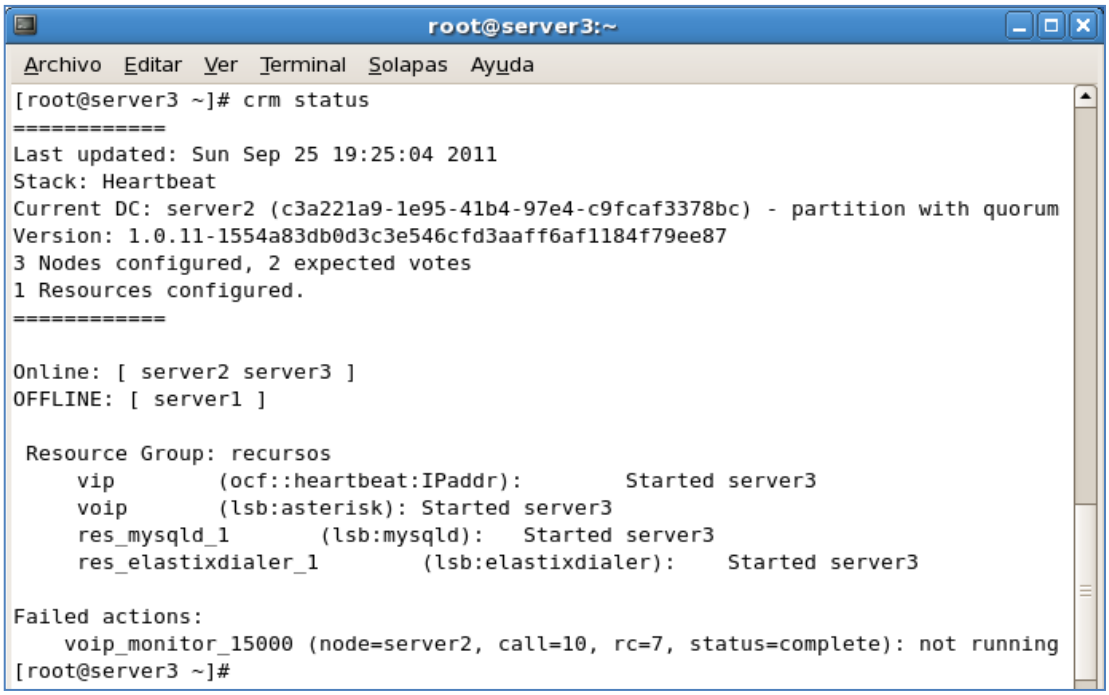

Figura V. 46 Registro de la caída del servicio asterisk

5.3. COMANDOS DE ADMINISTRACIÓN

Los recursos del clúster deben ser gestionados adecuadamente para que el administrador invierta la menor cantidad de tiempo en detectar, investigar y recuperar fallos de hardware y software, y de este modo definir posibles medidas de contingencia y tratar que el sistema esté libre de errores.

En la siguiente Tabla se recoge algunos comandos útiles en tareas de administración de recursos y en general de monitorización del clúster. Posteriormente se presentan capturas de estos comandos.

Tabla V. XII Comandos de administración del Clúster

COMANDO	DESCRIPCIÓN
crm status	Muestra información de cuantos nodos y recursos se encuentran configurados, el estado de los recursos y en que nodo se encuentran ejecutándose, así como también el estado de los nodos.
crm_mon -1f	Incluida la información que muestra el comando crm_status, presenta también el estado de la migración de los recursos, el valor de la cuenta de fallos y errores registrados.
crm node show	Información de los nodos del clúster, nombre, identificador, estado.
crm configure show	Muestra los parámetros de configuración establecidos en el clúster como un todo, es decir la información que se almacena en el archivo cib.xml
crm resource status	Informa el estado de los recursos del clúster, solo si están iniciados o no, mas no en que nodo se encuentran.
crm_verify -LV	Despliega los fallos que han ocurrido en el clúster.

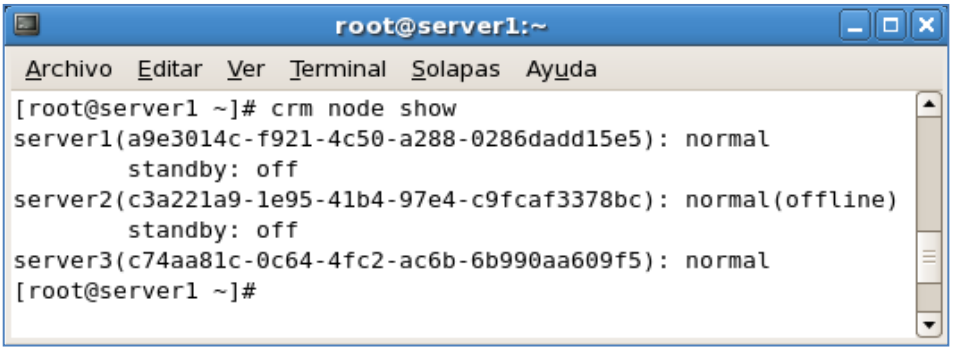

Figura V. 47 Comando **crm node show**

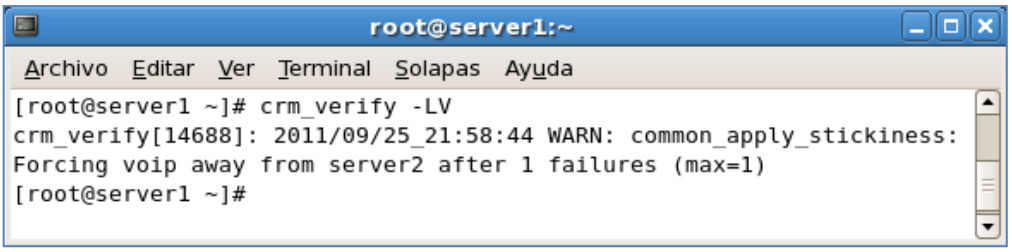

Figura V. 48 Comando **crm_verify -LV**

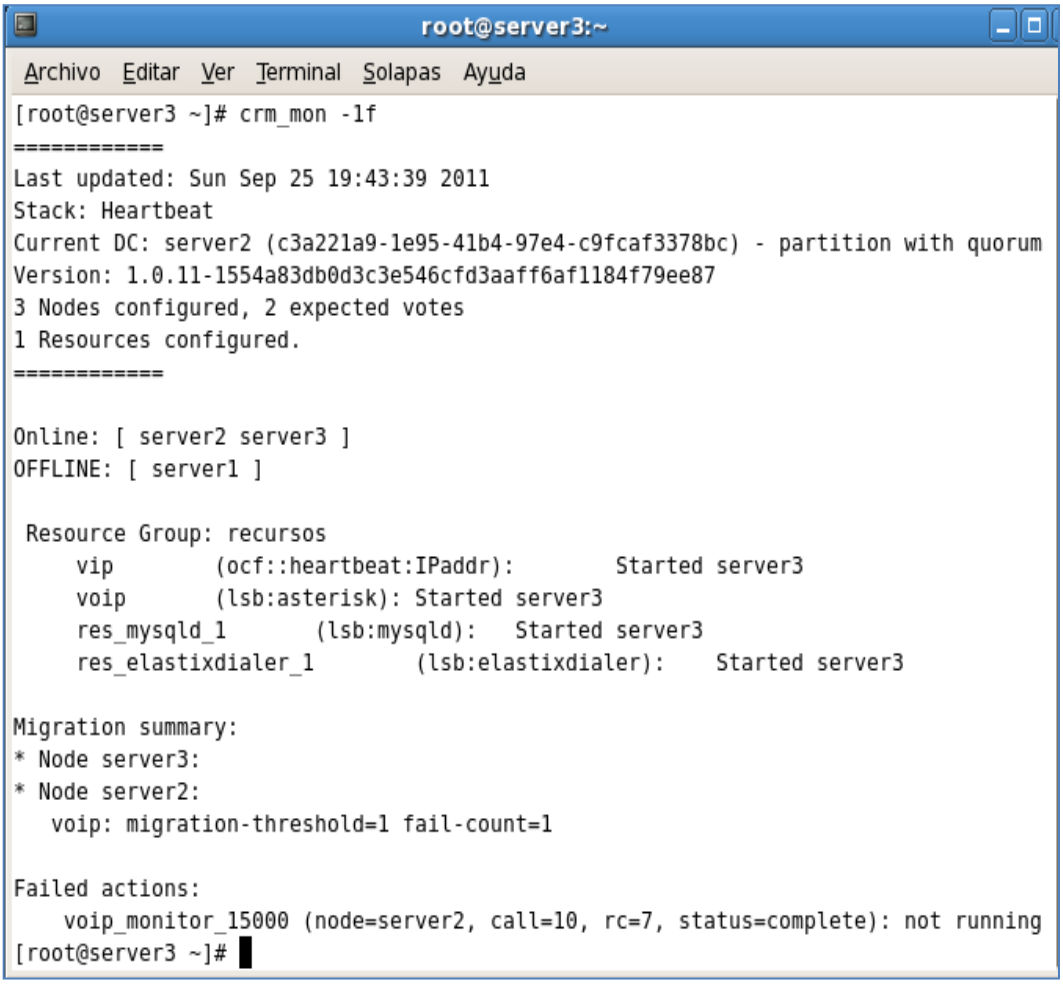

Figura V. 49 Comando **crm_mon -1f**

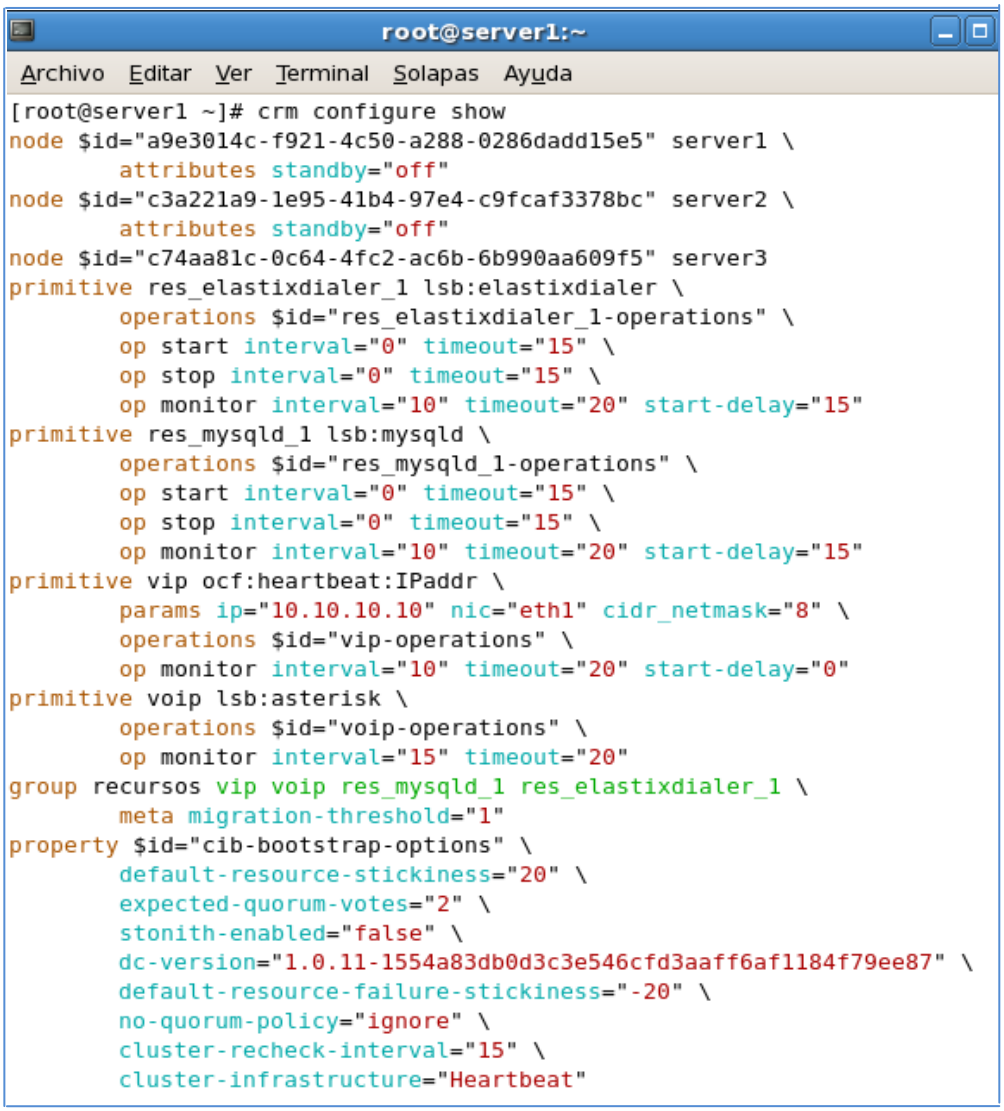

Figura V. 50 Comando **crm configure show**

La información antes mostrada a través del comando es posible también observarla desde la interfaz gráfica, siendo posible manipular estas variables con mayor facilidad.

En la Figura 51 se muestra una ventana de la interfaz gráfica, en el panel izquierdo esta la lista de los recursos configurados, en el centro se encuentra el grafico de los nodos y en el panel derecho se despliega la lista de variables que pueden ser modificadas según el administrador desee que actúe el clúster.

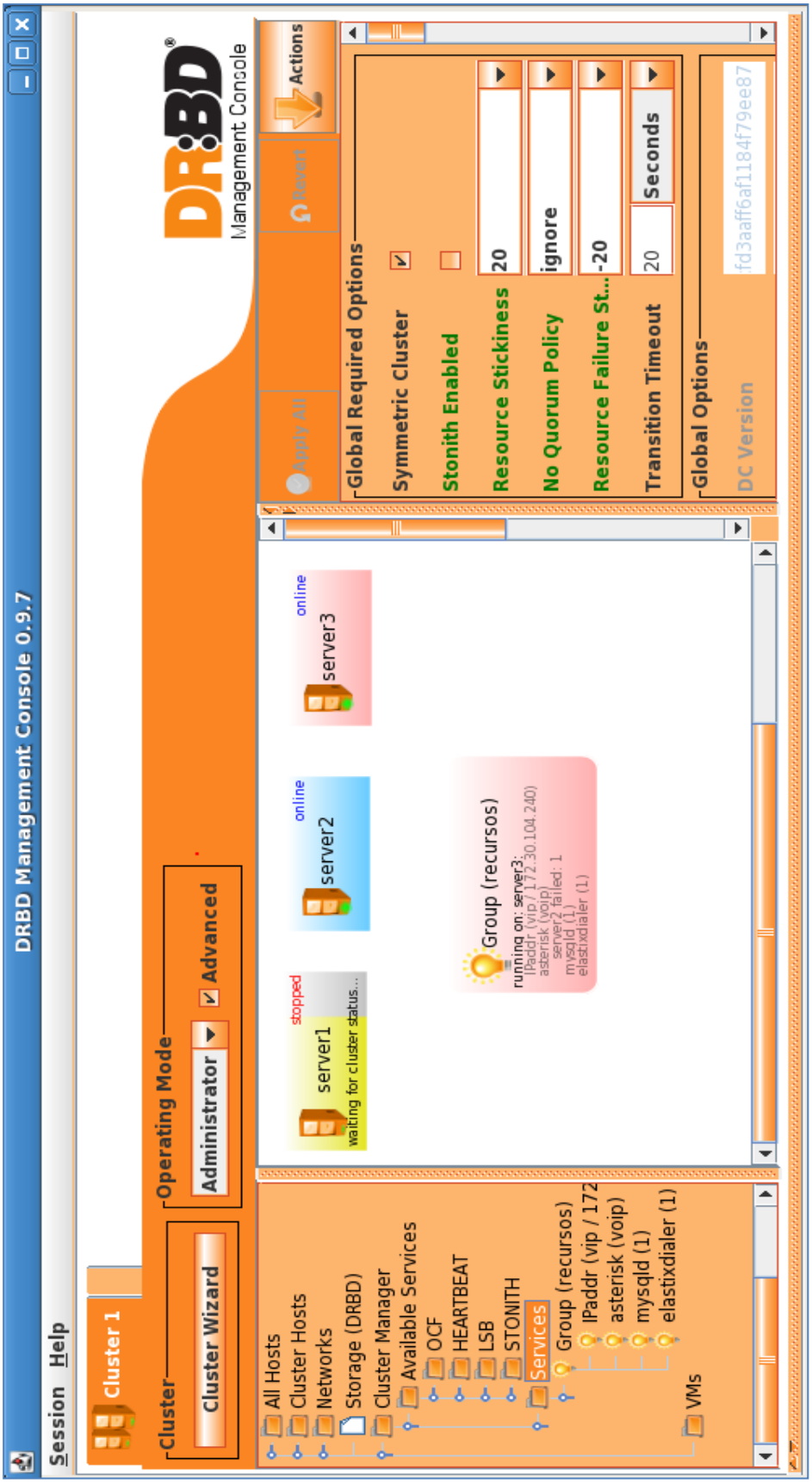

Figura V. 51 DMC

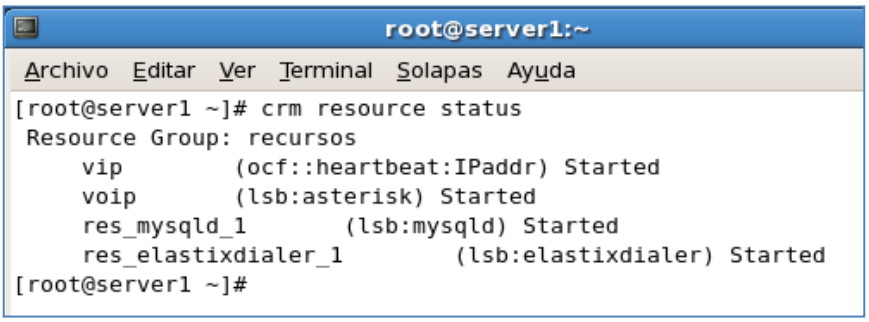

Figura V. 52 Comando **crm resource status**

5.4. COMPROBACIÓN DE LA HIPÓTESIS

Después de haber implementado un prototipo de Clúster de Alta Disponibilidad y haber realizado las pruebas de su comportamiento antes dos sucesos que provocan la migración de los recursos; se pudo recoger datos del tiempo en que el servicio no está disponible cuando ocurre un fallo, para posteriormente mediante el análisis estadístico de dichos datos poder demostrar si lo planteado en la hipótesis en cuanto a minimizar el tiempo de no accesibilidad a los Sistemas de Call Center, es verdad o no.

Debido a que el tiempo es un factor numérico y variante, es necesario utilizar un método estadístico denominado ANOVA (Análisis de la Varianza), cuyo principio básico es comparar la variación total de un conjunto de muestras para luego determinar el grado se significación de la hipótesis haciendo uso de la distribución F.

Los tiempos que se analizan son en base a escenarios que se los divide en dos grupos, tiempos de no accesibilidad en escenarios con clúster y en escenarios sin clúster, además para cada escenario se planea trabajar con 3 datos.

En escenarios con clúster, el tiempo mínimo en que el grupo de recursos tarda en estar activo en otro nodo, es de 45 segundos, pudiendo variar en 50 y 55 s. En cambio en escenarios sin clúster los tiempos de no accesibilidad van a depender de diferentes

situaciones, tales como, si se tiene otro servidor en el cual solo hace falta comenzar a levantar la infraestructura del Call Center, si se tiene un servidor pero sin ninguna instalación ni configuración, o si no se tiene ningún servidor y será necesario adquirirlo, en si el número de casos es extenso por lo que a modo general se toma en cuenta datos como: 3 horas, 1 día y 3 días.

Los valores antes mencionados se los expresa en horas y se los recoge en la siguiente tabla.

Tabla V. XIII Tiempos de no accesibilidad

	Con Clúster			**Sin Clúster**		
Escenario	**1**	**2**	**3**	**4**	**5**	**6**
Xi - tiempo(h)	0,0125	0,0139	0,0153	3	24	72

A continuación se detallan los pasos realizados para el Análisis de la Varianza del factor tiempo:

1) Se calcula la sumatoria de cuadrados de los tiempos de la tabla anterior, para lo cual se utilizará la media con la siguiente fórmula:

$$Media = \frac{\sum Xi}{n} = \frac{99{,}0417}{n} = 16{,}50695$$

Tabla V. XIV Sumatoria de Cuadrados Total

Xi	**Xi – Media**	$(\mathbf{Xi - Media})^2$
0,0125	-16,49445	272,066881
0,0139	-16,49305	272,020698
0,0153	-16,49165	271,97452
3	-13,50695	182,437698
24	7,49305	56,1457983
72	55,49305	3079,4786
		4234,12419=SCt

2) Cálculo de la sumatoria y la media para cada grupo, con el fin de obtener los efectos principales, es decir la diferencia de la media total con la media de cada grupo.

Tabla V. XV Efectos Principales

Xi	∑ del grupo	Media del grupo	Efecto Principal
0,0125			
0,0139			
0,0153	0,0417	0,0139	-16,49305
3			
24			
72	99	33	16,49305

3) Cálculo de la suma de cuadrados entre los grupos y dentro de los grupos:

$$SCe = \sum(Ygrupos - Ytotal)^2$$
$$SCe = 1632{,}12419$$

$$SCd = SCt - SCe$$
$$SCd = 2502$$

4) Cálculo de la varianza de los cuadrados medios

Varianza del cuadrado medio entre grupos=VCMe=SCe/(k-1)

Varianza del cuadrado medio dentro del grupo=VCMd=SCd/(n-k)

donde:

n(grupos)= valores de cada grupo=3

K= número de grupos=2

n(total)=6

(k-1)= grados de libertad entre grupos= 2-1= 1

(n-k)= grados de libertad dentro del grupo= 6-2= 4

$$VCMe = 1632{,}12419$$

$$VCMd = 625{,}500001$$

5) Cálculo del estadístico de la prueba F:

$$\boldsymbol{F = \frac{VCMe}{VCMd} = 2{,}60931125}$$

Los cálculos realizados se los resume en la siguiente tabla, denominada tabla ANOVA.

Tabla V. XVI Tabla ANOVA del factor tiempo

Origen de las Variaciones	SC	gl	VCM= SC/gl	F= VCMe/VCMd
Entre grupos (SCe)	1632,12419	1	1632,12419	**2,60931125**
Dentro de grupos (SCd)	2502	4	625,500001	
TOTAL (SCt)	4134,12419	5	826,824839	

6) Decisión de rechazo

Con el fin de saber el valor de significación de la hipótesis, se compara la F obtenida de los datos reales con la F de la tabla de Distribución de Fisher.

Para p= 0.05, se obtiene que **F= 6,61**

F(observada) < F(tabla), es decir **2,6093 < 6,61**

Por lo tanto se acepta la Hipótesis.

Además para cada tiempo de la Tabla V.2 se calculó su respectivo porcentaje de disponibilidad, con la siguiente fórmula:

$$\boldsymbol{Disponibilidad}(\%) = \frac{\boldsymbol{MTBF}}{\boldsymbol{MTBF + MTTR}} * \mathbf{100}$$

donde:

MTBF= tiempo entre fallos, expresado en horas.

MTTR= tiempo para reparar, expresado en horas.

Si se considera que para cada uno de los seis escenarios mencionados anteriormente, los fallos ocurrirían cada 15 días, se obtiene el valor correspondiente a MTBF, que expresado en horas sería igual a 360, y con los tiempos de no accesibilidad que corresponden a los tiempos para reparar (MTTR), se calculó la disponibilidad y se presenta a continuación.

Tabla V. XVII Porcentaje de Disponibilidad para escenarios con y sin Clúster

Escenario	**Con clúster**			**Sin Clúster**		
MTTR(h)	**0,0125**	0,0139	0,0153	3	24	72
D(%)	**99,997**	99,996	99,996	99,174	93,750	83,333

Es decir que en escenarios con clúster el tiempo de no accesibilidad si se minimiza, logrando un porcentaje de cuatro nueves, muy cercano al valor ideal de cinco nueves que se habló en capítulos anteriores.

ANÁLISIS FINAL

Considerando el estudio estadístico realizado y el cálculo de la disponibilidad, se comprueba que un Clúster de Alta Disponibilidad permite garantizar el funcionamiento del servicio que se brinda, evitando que fallos de hardware o software sean percibidos por parte de los usuarios, a la vez que también es útil su implementación en paradas del sistema programadas por motivos de mantenimiento de hardware o actualización de software.

CONCLUSIONES

- ✓ Un componente importante de un Call Center es la Tecnología de Integración COMPUTADOR-TELÉFONO que evita al agente de tareas repetitivas en cuanto a identificación de clientes y además permite realizar el registro de información relevante durante la ejecución de una campaña.

- ✓ Un ACD permite la distribución de la carga de llamadas a los agentes, haciendo uso de las colas de asterisk como mecanismo de gestión.

- ✓ Los CC se han convertido en la mejor herramienta para satisfacer necesidades de comunicación directa y personalizada entre empresa y cliente, no solamente en ambientes de negocios sino también en servicios de misión crítica.

- ✓ Al ser un CC una contribución para un servicio al cliente de excelencia, resulta indispensable disponer de estos de forma ininterrumpida, lo cual es posible lograr utilizando un Clúster de Alta Disponibilidad.

- ✓ El análisis de Alta disponibilidad mediante clustering ha permitido definir los procedimientos necesarios para llevar a cabo una configuración en la cual una caída del servicio de Call Center sea imperceptible para los usuarios.

- ✓ Una parte importante de un clúster de alta disponibilidad es el conjunto de componentes de software cuyos objetivos principales son los de monitorear constantemente los recursos del sistema, detectar fallos y efectuar tareas de auto-recuperación frente a estos.

- ✓ Un clúster es empleado para mejorar el rendimiento y/o la disponibilidad a niveles que un único sistema no puede alcanzar, o para ser una alternativa económica y equiparable a costosos sistemas de alta potencia y disponibilidad.

- ✓ La implementación del Prototipo ha puesto en evidencia los aspectos a ser considerados como mejores prácticas para el despliegue de un CC de alta disponibilidad en un ambiente de producción.

- ✓ El test de un Clúster de Alta disponibilidad mediante el apagado del nodo activo o caída del servicio, permitió simular fallos por cortes de energía, fallos en el hardware o caída completa del sistema operativo, posibilitando el monitoreo de tiempos de respuesta en la restauración del servicio y la forma de reacción del clúster frente a estos imprevistos.

RECOMENDACIONES

- ✓ Se deben definir los componentes a utilizar en la implementación de un Cal Center ya que dependiendo del servicio que se desee brindar sea este dedicado a atender llamadas (Campañas Entrantes) o a realizarlas (Campañas Salientes) se utilizara una configuración especifica de estos para cada escenario.

- ✓ Es aconsejable manejar a los recursos como un conjunto es decir crear un grupo de recursos ya que posibilita operaciones de administración a nivel general sobre su comportamiento.

- ✓ Tomar en cuenta qué recursos deben ser iniciados antes que otros, como es el caso de la dirección IP virtual y asterisk donde se requiere primero activa la VIP antes que asterisk pues este no sería accesible sin una VIP que permita alcanzar el servicio.

- ✓ Para evitar errores con las operaciones start/stop de los recursos configurados en el cluster, se los debe deshabilitar del arranque automático en el sistema operativo, para que sea heartbeat el encargado de controlar dichas operaciones en el momento de la migración.

- ✓ El sistema de almacenamiento para un clúster de alta disponibilidad debe ser seleccionado en base al número de nodos disponibles y a la configuración de las bases de datos que pueden estar reflejadas por seguridad, compartidas para aprovechar la capacidad de almacenamiento o sincronizadas para permitir la velocidad de recuperación del servicio.

- ✓ Se recomienda utilizar los comandos de administración para detectar fallas en el funcionamiento del clúster y poder definir posibles medidas de contingencia para que el sistema esté libre de errores.

RESUMEN

El Análisis e Implementación de Alta Disponibilidad mediante Clustering, en Sistemas de Call Center (CC) basados en Voz sobre el protocolo IP (VoIP), se realizó con el propósito de minimizar los tiempos de no accesibilidad a los servicios que brindan los mismos.

Se implementó un prototipo en ambiente virtual utilizando el emulador de computadores VMware Workstation, el escenario lo conformaron tres servidores en los cuales se instaló, el sistema operativo CentOS, Elastix para el ambiente VoIP, Heartbeat para la comunicación entre nodos, Pacemaker como Administrador de Recursos del Clúster y Rsync para la sincronización de los archivos.

El clúster fue sometido a sucesos que provocan la migración de los servicios para recopilar datos de los tiempos de recuperación del sistema, obteniendo que en escenarios con clúster el tiempo de no accesibilidad es de 45 segundos, lo que implica un porcentaje de disponibilidad del 99,997%, en cambio en escenarios sin clúster, los tiempos dependen de diferentes situaciones, tomando como ejemplo el caso en que se tarde 3 días en levantar el sistema completo se alcanza el 83,333% de disponibilidad.

Se determinó entonces que la utilización de Clústeres de Alta Disponibilidad permite obtener sistemas tolerantes a fallos, evitando que el mal funcionamiento de hardware o software afecte al servicio que se brinda y sobre todo permite minimizar la percepción de dichos fallos por parte de los usuarios.

SUMMARY

The High-Availability Analysis and Implementation through Clustering in Call Center Systems (CC) based on Voice over IP protocol (VoIP) was carried out to minimize the no-access services provided by them.

A prototype was implemented in the virtual environment using the emulator of computers VMware Workstation; the scenery was made up of three servers in which the operative system CentOS, Elastix for the VoIP environment, Heartbeat for the communication between nodes, Pacemaker as a Resources Administrator of the Cluster and Rsync for the file synchronization, were installed.

The cluster was subjected to events that cause migration of services to collect data of system recovery times, showing that in cluster scenery the no-access time is 45 seconds, which implies a high availability percentage of 99,997%; on the other hand, in no-cluster sceneries, the times depend on different situations; taking as an example, in the case in which it takes 3 days to survey the complete system 83,333% availability is reached.

It was determined that the use of High-Availability clusters permits to obtain systems tolerant to faults, avoiding the fact that the bad functioning of the hardware or software effects the provided service and, above all it permits to minimize the perception of such faults by the users.

GLOSARIO

Agente de Recurso.- Script que controla recursos de un clúster. Pueden ser de tipo LSB, OCF o Legacy.

Alta Disponibilidad.- Característica que tiene un sistema para protegerse o recuperarse de interrupciones o caídas, de forma automática y en un corto plazo de tiempo.

Call Center.- Conjunto tecnológico y administrativo que permite unificar la inteligencia y potencia de procesamiento de los sistemas informáticos y las facilidades de la conmutación de llamadas telefónicas, para suministrar información a los llamantes en un ambiente de intimidad personal

CIB.- Una representación de la configuración del clúster y el estado general (miembros del nodo, recursos, limitaciones, etc), escrito en XML y residente en la memoria.

Clúster.- Un clúster es un grupo de ordenadores, interconectados mediante una red, que trabajan conjuntamente y que se comportan como si fuesen un único sistema.

CRM.- Software encargado de verificar el estado de los recursos de un clúster, en caso de un fallo, reinicia automáticamente dichos recursos en otro nodo.

CTI (Computer Telephony Integration).- Software/Tecnología que permite al agente recibir simultáneamente la llamada y los datos del usuario que llama en su pantalla.

Failover.- Proceso por medio del cual un nodo que ha fallado, pasa el control a otro que está en espera de tomarlo.

Fencing.- Es la capacidad de un clúster para hacer que uno de sus nodos libere los recursos que tiene ocupados.

PBX.- Central Telefónica cuya misión es gestionar las extensiones telefónicas corporativas internas y conectarse a la Red Telefónica Pública Conmutada (RTPC-PSTN) para la comunicación con el exterior.

Recurso.- Cualquier tipo de servicio o aplicación que sea conocido por Heartbeat. Ejemplos incluyen una dirección IP, un sistema de archivos, una base de datos, etc.

SIP.- Protocolo que se encarga de iniciar, mantener y terminar sesiones multimedia, las cuales se llevan a cabo de manera interactiva. Por sesiones multimedia se refiere a aplicaciones de audio, video, mensajería instantánea, conferencias y aplicaciones similares.

Softphone.- Software que hace una simulación de teléfono convencional por computadora. Es decir, permite usar la computadora para hacer llamadas a otros softphones o a otros teléfonos convencionales.

Split-Brain.- Escenario en el que los nodos del clúster se dividen en dos o más grupos que no saben el uno del otro (ya sea a través de un fallo de software o hardware), provocando que más de un servidor o aplicación pertenecientes a un mismo clúster intenten acceder a los mismos recursos, lo que puede causar daños a dichos recursos.

Tiempo de inactividad.- Tiempo en que un sistema no se encuentra disponible para los usuarios.

VoIP.- Tecnología que permite la transmisión de voz a través de redes IP en forma de paquetes de datos.

BIBLIOGRAFÍA

1. **GONCALVES, F.E.** ASTERISK PBX Guía de la Configuración. Traducido del portugués por Oscar Fueyo. 1a. Ed. Janeiro, 2007. 362 p.

Bibliografía Internet

2. COMO INSTALAR HEARTBEAT
   http://www.linux-ha.org/doc/users-guide/users-guide.html
   [2011-07-09]

3. COMPONENTES DE UN CALL CENTER
   http://www.hostmyagents.com/features/acd-distribuidor-de-llamadas
   http://www.3cx.com/PBX/voicemail-system.html
   [2011-06-15]

4. CONCEPTOS DE HEARTBEAT, CLÚSTER-GLUE Y RESOURCE-AGENTS
   http://www.linux-ha.org/wiki/Main_Page
   [2011-06-30]

5. CONFIGURACIONES DE ALTA DISPONIBILIDAD
   http://www.lintips.com/?q=node/119
   [2011-06-22]

6. CONFIGURACION DE HEARTBEAT
   http://redes-privadas-virtuales.blogspot.com/2009/03/alta-disponibilidad-con-heartbeat.html
   http://systemadmin.es/2011/03/instalacion-y-configuracion-de-heartbeat-2
   [2011-07-13]

7. CONFIGURACIÓN DE PACEMAKER
http://denniscm20.blogspot.com/2010/04/alta-disponibilidad-en-linux-heartbeat.html
[2011-08-01]

8. DESCARGA DE ELASTIX
http://www.elastix.org/es/descargas.html
[2011-04-22]

9. ELEMENTOS DE UN AMBIENTE VoIP
http://es.wikipedia.org/wiki/Voz_sobre_Protocolo_de_Internet
[2011-05-03]

10. MANUAL DEL CALL CENTER
http://www.elastix.org/es/informacion-del-producto/manuales-libros.html
[2011-04-28]

11. MANUAL DE CONFIGURACIÓN DEL FICHERO RSYNCD.CONF
http://www.mediacollege.com/cgi-bin/man/page.cgi?section=5&topic=rsyncd.conf
[2011-09-05]

12. OPCIONES DE CONFIGURACIÓN DE RSYNC
http://ss64.com/bash/rsync.html
[2011-08-31]

13. PARTES DE UN MENSAJE SIP Y TRANSACCIONES SIP
http://www.voip.unam.mx/archivos/docs/Curso%20SIP_05012008.pdf
[2011-05-10]

14. PETICIONES Y RESPUESTAS SIP
http://www.voipforo.com/SIP/SIPmensajes.php
[2011-05-10]

15. SCRIPT PARA EL FUNCIONAMIENTO DE RSYNC
http://ral-arturo.blogspot.com/2011/05/i-cluster-basico-de-alta-disponibilidad.html
[2011-09-02]

16. TIPOS DE CLÚSTER
http://www.lintips.com/?q=node/117
[2011-06-21]

ANEXOS

Anexo #1

LISTA DE RESPUESTAS SIP

1xx = respuestas informativas

- 100 Tratando
- 180 Teléfono sonando
- 181 Llamada está siendo redireccionada
- 182 Encolada
- 183 Progreso de sesión

2xx = respuestas de éxito

- 200 OK
- 202 aceptada: Utilizada por referidos

3xx = respuestas de redirección

- 300 Múltiples opciones
- 301 Movido permanentemente
- 302 Movido temporalmente
- 305 Utiliza Proxy
- 380 Servicio alternativo

4xx = errores de solicitud

- 400 Solicitud errónea
- 401 No autorizado: Utilizado solamente por registradores. Proxys deben utilizar autorización proxy 407
- 402 Pago requerido (Reservado para uso futuro)
- 403 Prohibido
- 404 No Encontrado: Usuario no encontrado
- 405 Método no permitido
- 406 No Aceptable
- 407 Autenticación Proxy Requerida

- 408 Expiración de solicitud: No pudo encontrar al usuario a tiempo
- 410 Ido: El usuario existió una vez, pero ya no está disponible acá.
- 413 Solicitud de entidad muy larga
- 414 Solicitud URI muy larga
- 415 Tipo de medio no soportado
- 416 Esquema URI no soportado
- 420 Mala extensión: Mala extensión de protocolo SIP utilizada, no entendida por el servidor
- 421 Extensión requerida
- 423 Intervalo muy corto
- 480 Temporalmente no disponible
- 481 Llamada/Transacción no existe
- 482 Lazo detectado
- 483 Muchos saltos
- 484 Dirección incompleta
- 485 Ambiguo
- 486 Ocupado acá
- 487 Solicitud terminada
- 488 No aceptable acá
- 491 Solicitud pendiente
- 493 No descifrable: No pudo desencriptar la parte del cuerpo S/MIME

5xx = errores de servidor

- 500 Error interno del servidor
- 501 No Implementado: La solicitud / método SIP no está implementado acá
- 502 Pasarela errónea
- 503 Servicio no disponible
- 504 Expiración de servidor
- 505 Versión no soportada: El servidor no soporta esta versión del protocolo SIP
- 513 Mensaje demasiado largo

6xx = errores globales

- 600 Ocupado en todas partes
- 603 Declinación
- 604 No existe en ninguna parte
- 606 No Aceptable

Fuente: http://www.3cx.es/voip-sip/sip-responses.php